IDA

INSTITUTE FOR DEFENSE ANALYSES

JCATS Verification and Validation Report

W. M. Christenson, Project Leader
Mary Catherine Flythe
Terri J. Walsh
Robert A. Zirkle

October 2002

Approved for public release;
distribution unlimited.

IDA Document D-2791

Log: H 02-002230

This work was conducted under contract DASW01 98 C 0067, Task AJ-6-1543, for the Deputy Under Secretary of Defense for Advanced Systems and Concepts. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

INSTITUTE FOR DEFENSE ANALYSES

IDA Document D-2791

JCATS Verification and Validation Report

W. M. Christenson, Project Leader
Mary Catherine Flythe
Terri J. Walsh
Robert A. Zirkle

PREFACE

This document was prepared by the Institute for Defense Analyses in fulfillment of the Task Order entitled “Analysis Support for the Military Operations in Urban Terrain (MOUT) Advanced Concept Technology Demonstration (ACTD),” sponsored by Deputy Under Secretary of Defense for Advanced Technology (DUSD(AT)).

The authors greatly appreciate the helpful comments and assistance received during the development of this document. The authors would especially like to thank Dr. Dale Lichtblau of the Institute for Defense Analyses for reviewing this document, and the Constructive Simulation Center, Dismounted Battlespace Battlelab (DBBL) in Ft Benning, Georgia for their assistance in the validation.

CONTENTS

APPENDICES

ANNEXES

TABLES

EXECUTIVE SUMMARY

A. Introduction

This paper documents the Joint Conflict and Tactical Simulation (JCATS) verification and validation (V&V) effort performed by the Institute for Defense Analyses (IDA). The work was performed under task order AJ-6-1543: Analysis Support for the Military Operations in Urban Terrain (MOUT) Advanced Concept Technology Demonstration (ACTD) Extension.

Under previous tasking, IDA determined that, while few models even attempted to represent combat in urban areas and none fully represented all aspects of MOUT operations, Lawrence Livermore National Laboratory's (LLNL) JCATS model came closest to meeting the MOUT capability requirements. However, IDA further determined that before JCATS could be fully utilized for MOUT analysis purposes, both the model's urban combat representation and the relevant database needed to be subjected to appropriate verification and validation efforts. This report documents the results of an effort to undertake the first of these requirements: a V&V the model's representation of combat in the urban environment. [1]

Verification of the model involves determining that it accurately represents the developer's description and specifications; basically, that the model is performing as expected and stated by its developers. Validation of the model, on the other hand, is a check to determine whether it adequately represents a relevant slice of reality, in this case urban combat. The V&V of the JCATS model provides the basis for judgment on the part of managers and users with respect to acceptance or accreditation for an intended purpose; in this case, analyses addressing MOUT operations. This work builds on a previous JCATS V&V effort undertaken by the Non-Lethal Weapons Joint Program Office (JPO) and Fort Benning's Dismounted Battlespace Battle Laboratory (DBBL) to assess the model's use in analyses addressing non-lethal weapons issues. Although many of the same algorithms examined in this study were also assessed in the non-lethal weapon V&V, they were not looked at the context of MOUT operations.

[1] The task assigned to IDA was to V&V the urban portions of the JCATS model for analysis purposes. The database issue was not addressed, as it will largely vary on a study-by-study basis. The development of a general, broadly accepted database for urban operations has barely begun, and was outside the scope of this project. Other potential uses of the model – e.g., for training or planning purposes – also were not addressed in this project.

B. Methodology

For the JCATS V&V, we assessed the capabilities of JCATS for MOUT operations only; other types of operations using JCATS (e.g., littoral warfare, armored maneuver warfare) were not addressed unless they directly affected urban operations. One of the key differences distinguishing urban operations from other types of combat operations is the closed, complex terrain found in the former; terrain dominated in particular by the presence of buildings. Urban combat takes place in, around, and through buildings. Buildings extend ground combat to three dimensions, while blocking movement, and cutting down detection and engagement times. In addition, while other types of ground combat operations are conducted largely by tanks and other vehicles, MOUT operations are oftentimes principally the domain of the dismounted combatant. And the missions these forces are tasked to perform are often unique and complex: e.g., gaining access to a building, clearing and securing a building, and navigating through crowded streets.

1. Verification Methodology

To confirm that the model is performing as expected, (i.e., the verification portion of the V&V), we undertook both logical and code verification in accordance with recommended Army modeling methods and practices.[2] To further both verification efforts, we built on documentation review, code walk-through, algorithm checks, and peer review conducted during the course of the Non-Lethal Weapon JPO V&V effort. In addition, we developed and tested a series of vignettes designed to verify code execution. Specifically, we undertook the following activities:

- identified JCATS algorithms for MOUT relevance
- reviewed Non-lethal JPO V&V
- visited LLNL and reviewed the JCATS documentation to understand how the model developer's intended the JCATS functions to behave.
- developed vignettes for testing JCATS.

After identifying the relevant JCATS algorithms based upon MOUT requirements and reviewing the efforts of the Non-Lethal JPO verification work, we determined that the following algorithms remained to be examined in a MOUT context:

[2] Logical verification "is a review process to assure that the M&S algorithms correctly represent the intended processes in relation to the M&S requirements and specifications." Code verification is intended "to ensure that the representations of verified logic have been properly implemented in the computer code." See Headquarters, Department of the Army: *Verification, Validation, and Accreditation of Army Models and Simulation*, Department of the Army Pamphlet 5-11 (15 October 1993), pp. 6-8.

- Line of Sight (LOS)
- Line of Flight (LOF) of Auto Direct Fire
- LOF of Planned Direct Fire at Target
- LOF of Planned Direct Fire at Area
- LOF of Planned Indirect Fire
- LOF of Direct Support with Forward Observer (FO)
- LOF of Direct Support with Laser Designator (LD)
- Soldier Movement (including movement on ramps and through rubble, breaching, and entering buildings)
- Vehicle Blocking
- Miscellaneous (including algorithms encompassing various characteristics of ramps, fences, and stairs).

As part of the verification effort under the auspices of the Non-Lethal Weapon JPO, a line-by-line review of the JCATS computer code was undertaken of selected algorithms.[3] While leveraging off this effort, we choose a slightly different path for code verification: examining the model output from a set of tightly focused vignettes, each designed to test one or two specific algorithms. Within each vignette, in turn, we varied specific inputs in order to assess their impact on the model and determine whether it was operating as expected. We undertook our verification effort in this manner for several reasons. First, we did not feel the need to replicate many of the same activities already adequately undertaken under the Non-Lethal Weapon JPO effort. Second, we believed that conducting a verification effort in this manner allows for a broader check of the model's capabilities beyond a simple code review; for example, it includes a check of the manner in which data are cached and accessed as well as a test of whether subroutines are properly sequenced and accessed. In essence, we are considering these elements, as well as all the additional computer science "magic," as part of a "black box." Through a comparison of the inputs to and outputs from this box, we can assess the contents of the box itself: if the outputs appear reasonable and match expected results, given the inputs, we can conclude that the box is working as intended.

We developed 70 distinct vignettes, organized into 10 different sets, with each set designed to test a different MOUT-relevant JCATS algorithm. Each vignette was set up as a set of multiple shooter-target pairs, with each shooter-target pair reflecting a test condition of interest. For example, Vignette 1 was designed to test the effect of different

[3] Again, this review was undertaken with an eye towards the use of these algorithms in a non-lethal weapons context, rather than focusing on urban operations. Therefore, the review, while complete and adequate for its intended purpose, failed to address urban operations.

target postures (standing, crouching, prone, in a foxhole) or movements (crawling, walking, running) on the Line of Sight (LOS) algorithm and on the Line of Flight (LOF) Automatic Direct Fire algorithm. Vignette #1 uses seven shooter-target pairs. Separating the shooter-target pairs from each other by thin walls ensures that each shooter can only “acquire” his intended target. This setup allows – in this case – the simultaneous execution of seven sub-tests of the LOS and the LOF Auto Direct Fire algorithms without compromising the integrity of each particular sub-test. Using this scheme, we were able to devise a total of 424 separate sub-tests within the context of the 70 vignettes. The parameters examined in 424 sub-tests were selected based on several criteria: First, parameters were chosen based upon their presence in specific algorithmic equations. In some cases, preliminary tests were conducted to ensure that outcomes were independent of certain parameters. If found to be true, then these parameters were ignored in the remainder of the tests. For example, the first set of LOS tests indicated that the LOS was independent of seer’s posture; this parameter was ignored in the remainder of the tests. Brainstorming and discussions with subject matter experts were used to identify the most critical variables and parameters. These techniques were used to ensure a reasonable and appropriate set of parameter combinations were tested, based on the model’s equations and “real world” conditions.

During the course of IDA’s verification process, we worked closely with the modelers at LLNL, discussing problems encountered as well as potential solutions. Unlike many verification efforts, we were able to examine several successive evolutions of the JCATS model, each involving several improvements and enhancements. In this manner, we were able, in part, to check that previously identified problems had been corrected and identify any new ones that arose with each new release. We began verification testing of JCATS using version 2.3 of the model, moved to build 48 of version 3.0 (this was a Beta version), and ended up examining build 51.1 of version 3.0. Based on our investigations, a number of problems were found in version 2.3, and corrected in build 48 of version 3.0 of JCATS. When IDA received this version of the model, all 70 vignettes were retested. During this testing, additional problems were found, some of which were then corrected by LLNL in build 51.1, while others were still being worked on as of the completion of the IDA verification activities. IDA conducted final verification testing on build 51.1 to determine that fixes were made as indicated by LLNL. The results shown in this paper encompass the final results of the complete verification testing effort up through build 51.1 of version 3.0.

2. Validation Methodology

Validation is defined as the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended users. Understanding the difficulties involved in a full-fledged V&V of a force-on-force model, the goal of this effort was to determine the reasonableness of JCATS for MOUT representation to the greatest fidelity possible. Examining the output from the verification vignettes described above provided a partial solution as well to the validation effort: the model's output could be evaluated to determine how "realistic" was the model's behavior.

However, the bulk of the validation was accomplished by employing subject matter experts (SMEs) with knowledge of, and familiarity with, urban operations, who were asked to provide insights and judgments on how well JCATS represents "real" combat. These experts included individuals with considerable experience conducting and observing JCATS gaming activities, personnel with considerable experience conducting and observing urban training exercises, and soldiers who had been involved in actual U.S. military operations in urban environments. Personnel at the Constructive Simulation Center, Dismounted Battlespace Battlelab (DBBL) in Ft. Benning, Georgia, performed the actual validation work.

We began by isolating a set of key elements of urban combat, and representing these elements within a set of JCATS scenarios. Based on their real world knowledge and experience, the SMEs were asked to make judgments both on the operations as they witnessed them occurring on the JCATS screen as well as on the model's processed output. The following vignettes were chosen by assessment/review by the SMEs during the validation effort.

- Clear a floor
- Enter a building (breaching and entering 1st floor)
- Secure a street (outside operations)
- Attack a bunker or a strong point. Could call in artillery (if blocked, call for a Precision-Guided Mortar Munition (PGMM))
- Defend a building from attack

In developing the scenarios to be employed in this effort, DBBL chose to use the Objective Force Warrior (OFW) Situational Awareness (SA) force structure, with which it had recently conducted a series of JCATS model runs in support of the OFW program. Furthermore, arising from the work already performed under the OFW-SA study, DBBL had two different JCATS scenarios employing these forces readily available: one where blue forces were attacking into an urban area and one where they were defending urban

terrain. These two missions were selected because together they encompassed the six vignettes identified by IDA for the MOUT validation.

C. Results

1. Verification

Overall, the results of the verification strongly suggest that JCATS successfully demonstrated that its MOUT-associated algorithms performed as expected. Out of a total of 424 tests, 395 (over 93 percent) were judged to have passed; in other words, we determined that the test results in these cases were consistent with the intended behavior of the model. Again, we determined intended behavior based on JCATS documentation and discussions with the JCATS model developers at LLNL. Six of the nine algorithm groups passed all of their verification tests in version 3.0 of the model. The majority of the failures (19 out of 29) occurred during testing of the "Line of Flight – Direct Fire with Laser Designator" and were concerned, in whole or in part, with the fact that the model fails to check line of flight for laser-designated missiles. The remaining failures were of minor consequence, and none of the failures was judged to be "fatal." Errors were considered "fatal" if they caused the simulation to "crash," if they performed a calculation incorrectly, or if they involved a general or frequently conducted operation, task, or function found in MOUT operations. All of the errors were reported to LLNL and have been, or will be, addressed in later versions of the model. A total of eight tests within four of the vignettes could not be tested because it was not possible to set up the desired test; specifically, the model prohibited firing of munitions between floors.

2. Validation

The validation process should assess twenty-two Subject Matter Experts (SME's) were asked to judge whether JCATS provides a sufficient approximation to the real world. Twenty-one of these SME's were infantrymen with an average of sixteen years of military service. They were shown the replays of selected simulation production runs and excerpts from JCATS output files. They were given access to a JCATS client workstation and a qualified operator, and were permitted to "play" with the model.

To assess the model, each SME was asked to complete one or both parts of a two-part validation questionnaire. The first part of the questionnaire – addressing operational validation – was designed to determine whether JCATS output sufficiently represent the "real" world of urban combat. SME's with knowledge of, and familiarity with, urban operations were asked to complete this portion. The second part of the questionnaire –

addressing structural validation – was designed to assess whether the model's code, editors, and post-processing capabilities were adequate for representing the "real" world of urban combat (e.g., is the terrain resolution adequate for modeling urban operations). Those SME's with experience conducting and observing JCATS gaming, and with an in-depth understanding of the JCATS code, were asked to complete this portion of the questionnaire. Each question was to be answered on a one to five scale, with one meaning "not at all" and five meaning "very well."

The questionnaire scores – all averaging above 4.0 – suggest that the SME's strongly endorsed the view that both the operationally and structurally the JCATS model passed the validation test. In other words, the results suggest that the SME's felt that the representation of urban combat found in JCATS sufficiently and adequately represented the "real" world of MOUT operations. A similar result was found through a review the JCATS output derived from the verification testing.

D. Conclusions

Overall, we conclude that JCATS MOUT-related representations successfully passed both the verification and the validation examinations. The verification results strongly suggest that JCATS demonstrated that its MOUT-associated algorithms performed as expected. Likewise, the validation results strongly suggest that the model adequately represents the realities of combat in an urban environment. Again, this V&V of the JCATS model, combined with other efforts (e.g., the Non-Lethal Weapons JPO V&V), provides the basis for judgment on the part of managers and users with respect to acceptance or accreditation for a specific intended purpose: i.e., analyses addressing MOUT operations. Having presented the evidence, we will leave it to the relevant individuals to determine whether the model can be accredited for their particular study.

As with any large, constantly evolving model, the V&V of JCATS is an on-going process; not every element of the model has yet been reviewed (e.g., littoral warfare) and changes or additions to the model occur regularly. On the latter point, one caveat should be noted: Since this V&V was completed, a minor modification to the model has been released (version 3.1). The latest version of JCATS (version 4.0) was released in October 2002. Major changes in 4.0 include a new detection model (ACQUIRE) and the addition of nuclear weapons. Few, if any, changes, however, were made to the specific algorithms examined in this V&V study. Future users of the model, nonetheless, may want to check any minor modifications made to these algorithms in a MOUT context, as well as the

major changes and additions made to other algorithms in 4.0, prior to their accreditation of JCATS for analyses entailing urban operations.

MAIN REPORT

A. Introduction

This paper documents the Joint Conflict and Tactical Simulation (JCATS) verification and validation (V&V) effort performed by the Institute for Defense Analyses (IDA). The work was performed under task order AJ-6-1543: Analysis Support for the Military Operations in Urban Terrain (MOUT) Advanced Concept Technology Demonstration (ACTD) Extension.

Under previous tasking, IDA determined that, while few models even attempted to represent combat in urban areas and none fully represented all aspects of MOUT operations, Lawrence Livermore National Laboratory's (LLNL) JCATS model came closest to meeting the MOUT capability requirements. However, IDA further determined that before JCATS could be fully utilized for MOUT analysis purposes, both the model's urban combat representation and the relevant database needed to be subjected to appropriate verification and validation efforts. This report documents the results of a project to V&V the model's representation of combat in the urban environment.[1]

Verification of the model involves determining that it accurately represents the developer's description and specifications; basically, that the model is performing as expected and stated by its developers. Validation of the model, on the other hand, is a check to determine whether it adequately represents a relevant slice of reality, in this case urban combat.[2] The V&V of the JCATS model provides the basis for judgment on the part of managers and users with respect to acceptance or accreditation for an intended purpose; in this case, analyses addressing MOUT operations. This work builds on a previous JCATS V&V effort undertaken by the Non-Lethal Weapons Joint Program Office (JPO) and Fort Benning's Dismounted Battlespace Battle Laboratory (DBBL) to assess the model's use in analyses addressing non-lethal weapons issues. Although many of the same algorithms examined in this study were also assessed in the non-lethal weapon V&V, they were not looked at the context of MOUT operations.

[1] The task assigned to IDA was to V&V the urban portions of the JCATS model for analysis purposes. The database issue was not addressed as it will largely vary on a study by study basis. The development of a general, broadly accepted database for urban operations has barely begun, and was outside the scope of this project. Other potential uses of the model – e.g., for training or planning purposes – also were not addressed in this project.

[2] All models abstract from reality, and therefore none fully represents all the myriad elements and complexities of the real world. The issue for validation is whether or not the model adequately represents enough of the real world, with sufficient fidelity, to be useful for analyses addressing a specific portion of that reality.

The remainder of the paper is organized as follows: After this introduction is a description of the methodologies used in our V&V of JCATS. We then summarize the results of both the verification and the validation of JCATS. The main body of the report then concludes with a general summary of the V&V results. The report includes two annexes and ten appendices. Annex A contains our list of the algorithms and identifies those algorithms studied under the Non-Lethal JPO V&V. Annex B lists a set of MOUT modeling capabilities and requirements based on an assessment performed by IDA under a separate task undertaken for the Joint Staff. The annex also matches up JCATS capabilities with these requirements and describes any special features or limitations with JCATS with respect to a specific requirement. Appendix A is the original JCATS V&V Plan for MOUT. Although particular elements of the plan were modified as the study progressed, this appendix provides a broad outline of the process and a detailed justification for the general approach that we took in our V&V efforts. Appendix B contains detailed notes and diagrams on the operation of JCATS based on the September 2000 meeting of IDA personnel with LLNL JCATS personnel. It presents, in part, a description of the model developers' intended functions and capabilities of the various algorithms examined in this V&V effort. This appendix also serves as a tutorial for those individuals unfamiliar with the operation of JCATS. Appendix C describes the six different fire missions used by the JCATS model and examined in the verification portion of this V&V report. Appendix D contains detailed descriptions of the test vignettes used for verification, along with a summary of the test results and the problems encountered during testing. The details of each vignette test are contained in Appendix E. These details include the specifics on the setup, the results, and the pass/fail status of each part of a vignette. Appendix F contains further explanations of the problems found during the verification testing. Appendix G contains summaries of our email correspondence with LLNL describing the problems or questions encountered during the verification testing and the responses or resolutions to these issues. For future users of the model, Appendix H provides suggestions for work-arounds to problems we encountered during the verification testing. Appendix I contains a description of proposed changes to JCATS based on our verification testing, while Appendix J describes a list of previously proposed changes made by IDA on behalf of the MOUT ACTD. Appendix K contain descriptions of the scenarios used in the validation effort, while Appendix L lists the questions answered by the SMEs during the validation. The report concludes with a list of the acronyms used throughout the document.

B. Methodology

For the JCATS V&V, we assessed the capabilities of JCATS for MOUT operations only; other types of operations using JCATS (e.g., littoral warfare, armored maneuver warfare) were not addressed unless they directly affected urban operations. One of the key differences distinguishing urban operations from other types of combat operations is the closed, complex terrain found in the former; terrain dominated in particular by the presence of buildings. Urban combat takes place in, around, and through buildings. Buildings extend ground combat to three dimensions, while blocking movement, and cutting down detection and engagement times. While many other types of ground combat operations are conducted largely by tanks and other vehicles, MOUT operations are oftentimes principally the domain of the dismounted combatant. And the missions these forces are tasked to perform are often unique and complex: e.g., gaining access to a building, clearing and securing a building, and navigating through crowded streets.

1. Verification Methodology

To confirm that the model is performing as expected, (i.e., the verification portion of the V&V), we undertook both logical and code verification in accordance with recommended Army modeling methods and practices.[3] To further both verification efforts, we built on documentation review, code walk-through, algorithm checks and peer review conducted during the course of the Non-Lethal Weapon JPO V&V effort. In addition, we developed and tested a series of vignettes designed to verify code execution. Specifically, we undertook the following activities:

- identified JCATS algorithms for MOUT relevance
- reviewed Non-lethal JPO V&V
- visited LLNL and reviewed the JCATS documentation to understand how the model developer's intended the JCATS functions to behave.
- developed vignettes for testing JCATS.

After identifying the relevant JCATS algorithms based upon MOUT requirements and reviewing the efforts of the Non-Lethal JPO verification work, we determined that the following algorithms remained to be examined in a MOUT context:

---

[3] Logical verification "is a review process to assure that the M&S algorithms correctly represent the intended processes in relation to the M&S requirements and specifications." Code verification is intended "to ensure that the representations of verified logic have been properly implemented in the computer code." See Headquarters, Department of the Army: *Verification, Validation, and Accreditation of Army Models and Simulation*, Department of the Army Pamphlet 5-11 (15 October 1993), pp. 6-8.

- Line of Sight (LOS)
- Line of Flight (LOF) of Auto Direct Fire
- LOF of Planned Direct Fire at Target
- LOF of Planned Direct Fire at Area
- LOF of Planned Indirect Fire
- LOF of Direct Support with Forward Observer (FO)
- LOF of Direct Support with Laser Designator (LD)
- Soldier Movement (including ramps, breaching, rubble, entering buildings)
- Vehicle Blocking
- Miscellaneous (including bullet-proof glass workaround).

As part of the verification effort under the auspices of the Non-Lethal Weapon JPO, a line-by-line review of the JCATS computer code was undertaken of selected algorithms.[4] While leveraging off this effort, we choose a slightly different path for code verification: examining the model output from a set of tightly focused vignettes, each designed to test one or two specific algorithms. Within each vignette, in turn, we varied specific inputs in order to assess their impact on the model and determine whether it was operating as expected. We undertook our verification effort in this manner for several reasons. First, we did not feel the need to simply replicate many of the same activities already adequately undertaken under the Non-Lethal JPO effort. Second, we believed that conducting a verification effort in this manner allows for a broader check of the model's capabilities beyond a simple code review; for example, it includes a check of the manner in which data are cached and accessed as well as a test of whether subroutines are properly sequenced and accessed. In essence, we are considering these elements, as well as all the additional computer science "magic," as part of a "black box." Through a comparison of the inputs to and outputs from this box we can assess the contents of the box itself: if the outputs appear reasonable, given the inputs, then we can conclude that the box is working as intended.

We developed 70 distinct vignettes, organized into 10 different sets, with each set designed to test a different MOUT-relevant JCATS algorithm. Each vignette was set up as a set of multiple shooter-target pairs, with each shooter-target pair reflecting a test condition of interest. For example, Vignette 1 was designed to test the effect of different target postures (standing, crouching, prone, in a foxhole) or movements (crawling, walking, running) on the Line of Sight (LOS) algorithm and on the Line of Flight (LOF)

[4] Again, this review was undertaken with an eye towards the use of these algorithms in a non-lethal weapons context, rather than focusing on urban operations. Therefore, the review, while complete and adequate for its intended purpose, failed to address urban operations.

Automatic Direct Fire algorithm. The Vignette #1 uses seven shooter-target pairs. Separating the shooter-target pairs from each other by thin walls ensures that each shooter can only "acquire" his intended target. This setup allows – in this case – the simultaneous execution of seven sub-tests of the LOS and the LOF Auto Direct Fire algorithms without compromising the integrity of each particular sub-test. Using this scheme, we were able to devise a total of 424 separate sub-tests within the context of the 70 vignettes. The parameters examined in 424 sub-tests were selected based on several criteria: First, parameters were chosen based upon their presence in specific algorithmic equations. In some cases, preliminary tests were conducted to ensure that outcomes were independent of certain parameters. If found to be true, then these parameters were ignored in the remainder of the tests. For example, the first set of LOS tests indicated that the LOS was independent of seer's posture; this parameter was ignored in the remainder of the tests. Brainstorming and discussions with subject matter experts were used to identify the most critical variables and parameters. These techniques were used to ensure a reasonable and appropriate set of parameter combinations were tested based on the model's equations and "real world" conditions.

During the course of IDA's verification process, we worked closely with the modelers at LLNL, discussing problems encountered as well as potential solutions. Unlike many verification efforts, we were able to examine several successive evolutions of the JCATS model, each involving several improvements and enhancements to the model. In this manner, we were able, in part, to check that previously identified problems had been corrected and to identify any new ones that arose with each new release. We began verification testing of JCATS using version 2.3 of the model, moved to build 48 of version 3.0 (this was a Beta version), and ended up examining build 51.1 of version 3.0. Based on our investigations, a number of problems were found in version 2.3, and corrected in build 48 of version 3.0 of JCATS. When IDA received this version of the model, all 70 vignettes were retested. During this testing, additional problems were found, some of which were then corrected by LLNL in build 51.1, while others were still being worked on as of the completion of the IDA verification activities. IDA conducted final verification testing on build 51.1 to determine that fixes were made as indicated by LLNL. The results shown in this paper encompass the final results of the complete verification testing effort up through build 51.1 of version 3.0.

Appendix A contains a list of the prioritized algorithms and identifies those algorithms studied under the Non-Lethal JPO V&V. Appendix B contains detailed notes and diagrams based on the September 2000 meeting of IDA personnel with LLNL

JCATS personnel. Appendix D contains a summary descriptions of the vignettes along with the results of the runs performed using these vignettes and the problems encountered during testing. This appendix also identifies those vignettes that failed to pass their respective test. The details of each vignette are contained in Appendix E. The details include the specifics on the setup, the results, and the pass/fail status of each part of a vignette.

2. Validation Methodology

Validation is defined as the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended users.[5.] Understanding the difficulties involved in a full-fledged V&V of a force-on-force model, the goal of this effort was to determine the reasonableness of JCATS for MOUT representation to the greatest fidelity possible.[6] Examining the output from the verification vignettes described above provided a partial solution as well to the validation effort: the model's output could be evaluated to determine how "realistic" was the model's behavior. Problems or inconsistencies in the validation realm are pointed out in the verification section below.

However, the bulk of the validation was accomplished by employing subject matter experts (SMEs) with knowledge of, and familiarity with, urban operations, who were asked to provide insights and judgments on how well JCATS represents "real" combat. These experts included individuals with considerable experience conducting and observing JCATS gaming, such as the personnel at Fort Benning Simulation Center, and individuals with considerable experience conducting and observing urban training exercises and who have also been involved in actual U.S. military operations in urban environments. We began by isolating a set of key elements of urban combat, and representing these elements within a set of JCATS scenarios. The SMEs, based on their real world knowledge and experience, were then asked to make judgments both on the operations as they witnessed them occurring on the JCATS' screen as well as on the model's processed output. The following scenarios were assessed/reviewed by the SMEs during the validation effort.

- Clear a floor
- Enter a building (breaching and entering 1st floor)

---

[5] Reference: DODD 5000.59.

[6] For a more detailed discussion and justification of the methodology employed in the validation portion of this effort, see the V&V Plan, Appendix A.

- Secure a street (outside operations)
- Attack a bunker or a strong point. Could call in artillery (if blocked, call for a Precision-Guided Mortar Munition (PGMM))
- Defend a building from attack

Appendix K contains a more detailed discussion of each of the scenarios. After reviewing each scenario, SMEs were asked to complete a questionnaire. The questionnaires are contained in Appendix L.

C. Results

1. Verification

Table 1 summarizes the final results of the Verification testing (i.e., for version 3.0, build 51.1). Overall, the results strongly suggest that JCATS successfully demonstrated that its MOUT-associated algorithms performed as expected. Out of a total of 424 tests, 395 (over 93 percent) were judged to have passed; in other words, we determined that the test results were consistent with the intended behavior of the model. Again, we determined intended behavior based on JCATS documentation and discussions with the JCATS model developers at LLNL. The majority of the failures (19 out of 29) occurred during testing of the "Line of Flight – Direct Fire with Laser Designator," and were concerned, in whole or in part, with the fact that the model fails to check line of flight for laser designated missiles.

As problems were encountered during the setup and testing of the vignette's they were reported to LLNL. Appendix F contains a running log of the problems found, the scenario in which the problem occurred, LLNL's response to the stated problem, and the current status of resolution of the problem. The reported problems ranged in severity from benign (such as corrections or changes to the JCATS documentation) to moderate (coding errors affecting a narrow range of capabilities). Often, the same problem would occur in several vignettes in the same general testing area. Usually, the problem could be narrowed down to a single error in an algorithm. In a few cases, the purported problem turned out not to be error, but simply an issue that required further clarification. No fatal errors were detected in any of the tests. Errors were considered "fatal" if they caused the simulation to "crash," if they performed a calculation incorrectly, or if they involved a general or frequently conducted operation, task, or function found in MOUT operations. A total of eight tests within four of the vignettes could not be tested because it was not possible to set up the desired test; specifically, firing of munitions between floors was prohibited by the model.

During the verification process there were some problems encountered that were not associated with the testing of a specific vignette but were encountered in the general setup of tests. This group of problems relate mainly to the operation of the Terrain Editor and the Simulation module. Some of these problems can be attributed to the conversion from the HP-based version 2.3 to the PC-based version 3.0. The conversion required substantial restructuring of JCATS, and LLNL is still debugging some of the modules, particularly the Terrain Editor. Most of these problems are currently being resolved by LLNL.

Table 1. Summary of Verification Vignette Test Results

Algorithms	# of Vignettes/ Tests	Passed/ Not Tested	Failed	
LOS	12/71	71/0		
LOF - Auto Direct Fire by Soldiers	7/60	58/2		
LOF - PDF Soldier at Soldier	9/48	46/2		
LOF - PDF at Area with Soldiers	9/48	48/0		
LOF - Planned Indirect Fire	7/34	28/2	4	
LOF - Auto Indirect Fire with FO	7/34	27/2	5	
LOF - Direct Fire with LD	7/34	15/0	19	
Soldier Movement	8/74	73/0	1	
Vehicle Blocking	2/14	14/0		
Miscellaneous	7/7	7/0		
Total	**424**	**395**	**29**	
		93.2%	**6.8%**	

The following subsections describe the Verification results by algorithm category.

a. Line of Sight (LOS)

As defined in the JCATS documentation, Line of Sight (LOS) is a direct line from the viewer to the entity being viewed. If this is unobstructed by terrain, terrain features, smoke, or vehicles, then a line of sight can be established, and the target acquisition process may begin.[7] Two factors are used in determining Line of Sight: attenuation and exposure. Attenuation involves the reduction of visibility (or signal strength) due to intervening vegetation or smoke. Exposure accounts for the reduction in the amount of the target viewable due to the partial blockage of LOS by vehicles, terrain, buildings, and other features.[8]

The LOS algorithm determines (1) whether the view from the entity's sensor to a potential target is physically blocked, and (2) the amount of exposure of the target to the viewing entity.[9] To determine whether LOS is blocked, the model draws a ray from the sensor to the top of the target and another ray from the sensor to the foot of the target. If the top ray is blocked, i.e., the line of sight from the viewer to the top of the target is blocked, then the model determines that there is no LOS. If the top is not blocked, the model checks for objects blocking the foot ray. If the foot ray is blocked, this ray is raised until there is no blockage. The resulting portion of the target subtended by the angle between these two rays is the exposure height of the target. If the exposed height of the target is 0, there is no LOS to the target.

For Automatic Direct Fire missions[10], LOS implies a clear Line of Flight (LOF); i.e., if the line of sight is unblocked and the shooter successfully detects (acquires) the target, he will fire and the model assumes that the munition can reach the target. Alternatively, if there is no LOS, then no firing engagement occurs. If the target can be seen, the exposed height of the target is used in determining the expected Probability of Hit (PH). [11] Under these conditions, LOS impacts acquisition and the munition's expected PH, allowing the LOS tests to examine of elements these capabilities as well.

---

[7] JCATS Algorithm Manual, LLNL, Version 2.0 Draft, 30 September 1999, UCRL-MA-135117 DR, p 2-1.

[8] JCATS Algorithm Manual, LLNL, Version 2.0 Draft, 30 September 1999, UCRL-MA-135117 DR, p 2-10.

[9] Note that LOS is determined independent of the capabilities of the sensor.

[10] Automatic in the sense that the model determines whether an engagement occurs without additional user input. A Direct Fire mission is one that relies on probability of hit and probability of kill data input to determine the outcome of the engagement.

[11] Within the JCATS model, two categories of PH are determined: first, the expected PH which is calculated by the model given relevant inputs; and second, the resulting PH, which is determined by a Monte Carlo random draw at the time of a direct-fire event.

The LOS algorithm was tested using soldiers with M16 rifles firing Direct Fire missions at enemy soldiers under a variety of LOS conditions: e.g., different JCATS object classes (terrain, vegetation, fence, wall window), of various heights, placed between the shooter and target.

Various LOS vignettes are shown in Table 2. The vignettes were designed to test the LOS algorithm to determine if the target could be seen and the amount of exposure of the target. JCATS outputs were evaluated using the following measures:

- Whether the target was successfully acquired by the shooter, and
- The expected PH value when the shooter fired his weapon.

Table 2. LOS Vignettes

Vignette ID	Vignette Test	Setup Notes
VV01	target posture: seer standing, target in 7 postures	Use M16 rifle. Seven pairs of seer/target separated by thin buildings.
VV02	target posture: seer crouching, target in 7 postures	Use M16 rifle. Seven pairs of seer/target separated by thin buildings.
VV03	target posture: seer in pop-up, target in 7 postures	Use M16 rifle. Seven pairs of seer/target separated by thin buildings.
VV04	defilade: seer standing, target in partial or full defilade	Use M16 rifle. Two pairs of seer/target separated by thin buildings.
VV05	full blockage: seer & target standing, 4 types of blockage	Use M16 rifle. Four pairs of seer/target separated by thin buildings.
VV06	multiple partial blockage: seer & target standing, 4 types of blockage in 6 combinations	Use M16 rifle. Six pairs of seer/target separated by thin buildings.
VV06b	multiple partial blockage: seer & target standing, 4 types of blockage in 6 combinations	Use M16 rifle. Six pairs of seer/target separated by thin buildings.
VV06c	multiple partial blockage: seer & target standing, 4 types of blockage in 6 combinations	Use M16 rifle. Six pairs of seer/target separated by thin buildings.
VV07	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use M16 rifle. Five pairs of seer/target separated by thin buildings.
VV08	full target in window is partially blocked by objects: seer & target standing, 10 combinations of 4 types of partial blockage	Use M16 rifle. Ten pairs of seer/target separated by thin buildings.
VV09	partial target in window is blocked by objects: seer & target standing, 5 combinations of 4 types of partial blockage	Use M16 rifle. Five pairs of seer/target separated by thin buildings.
VV09a	partial target in window is blocked by objects: seer & target standing, 5 combinations of 4 types of partial blockage	Use M16 rifle. Five pairs of seer/target separated by thin buildings.

When testing version 2.3, several severe problems were identified with the LOS algorithm. These errors have been corrected in version 3.0, and the algorithm now passes the verification tests: i.e., the results obtained were those expected from the model based

on the intent of the developers. Specifically, we found the following results while testing the LOS algorithm:

- LOS is blocked by terrain, fences, exterior walls and doors, and interior walls and doors, assuming they are opaque.
- LOS is NOT blocked by fences, exterior walls and doors, and interior walls and doors if they are clear (i.e., Probability of Line of Sight Blockage, or PLOSB = 0.0). Note that terrain is always opaque.
- LOS is attenuated by vegetation based on its user-inputted PLOSB value; however, vegetation cannot partially block LOS.[12]
- LOS can be partially blocked by terrain, fences, and exterior walls with windows.
- LOS is not obtained if the target that is seen through a window has his head above the window (i.e., if his head is not exposed).
- The amount of exposure of a target affects the expected Probability of Hit (PH) of PHPK munitions.[13]

The fact that LOS is NOT blocked by non-opaque fences, exterior walls and doors, and interior walls and doors does present a minor validation problem: munitions may be blocked by these objects under certain conditions, but not under others. Specifically, munitions fired under automatic Direct Fire missions would not be block by these objects, because for these missions "LOS implies LOF." Under all other missions, however, the munitions would be blocked by the objects. For a further discussion of this situation, see Problem # 20 in Appendix F.

b. Line of Flight (LOF) of Auto Direct Fire

The "LOF of Auto Direct Fire" algorithm was tested using soldiers with M16 rifles firing at enemy soldiers under automatic Direct Fire missions, again under various LOF conditions (similar to the LOS conditions). As was the case for LOS, LOF for automatic Direct Fire missions impacts the expected PH calculations allowing these LOF tests to examine elements of the PH algorithm as well. Table 3 presents a series of LOF of Auto Direct Fire vignettes.

[12] Attenuation affects the target's signal strength as received by the shooter's sensor, and hence whether a target is acquired by the shooter; while blockage implies that a portion of the target is unseen and protected, and hence affects expected PH of the shooter's munition.

[13] PHPK munitions are those munitions whose effects on targets are evaluated using JCATS' PHPK assessment methodology. For more on this methodology see Appendix C.

Table 3. Line of Flight (LOF) of Auto Direct Fire Vignettes

Vignette ID	Vignette Test	Setup Notes
VV10a	full blockage: seer & target standing, 6 types of blockage using opaque objects	Use M16 rifle. Six pairs of seer/target separated by thin buildings.
VV10b	full blockage: seer & target standing, 6 types of blockage using clear objects	Use M16 rifle. Six pairs of seer/target separated by thin buildings.
VV10c	full blockage: seer & target standing, 6 types of blockage; auto and planned direct fire testing inside buildings	Use M16 rifle. Six pairs of seer/target separated by thin buildings.
VV11	multiple blockage: seer & target standing, 6 types of blockage in 6 combinations	Use M16 rifle. Six pairs of seer/target separated by thin buildings.
VV12	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use M16 rifle. Five pairs of seer/target separated by thin buildings.
VV13	flight through window is blocked: seer & target standing, 6 types of blockage	Use M16 rifle. Six pairs of seer/target separated by thin buildings.
VV14	fight through floors & ceilings: seer & target standing, 2 cases	Use M16 rifle. Two pairs of seer/target in separate buildings.

These tests were set up with “Shoot” on and “Hold Fire” off, so that the shooter will shoot when the target is acquired.

JCATS outputs were evaluated based on the following measures:

- Whether the target was successfully acquired by the shooter,
- The expected PH value when the shooter fired his weapon, and
- The effects, if any, of the munition on the target.

While testing version 2.3, the same problems reported in the LOS algorithm also affected the results of Line of Fire (LOF) of Auto Direct Fire testing. Again, these problems were fixed in version 3.0, and the tests for the LOF of Auto Direct Fire algorithm passed the verification tests. The following results were recorded while testing the LOF of Auto Direct Fire algorithm:

- LOS implies LOF; i.e., if there is LOS, then the weapon is fired in Auto Direct Fire and LOF is assumed not to be blocked
- If LOS is blocked completely, then there is no LOF and the weapon is not fired.
- If LOS is partially blocked by terrain, fences, or exterior walls with windows, then a portion of the target is not seen and hence is protected. The target’s exposed height is used to determine the expected PH of the shooter’s munition. However, the weapon is fired and LOF is not blocked.
- LOS cannot be partially blocked by vegetation, therefore LOF is not blocked and the weapons is fired in this case.

- If the target, as seen through a window has his head above the window height, LOS is not obtained and the weapon is not fired.
- The amount of exposure of a target affects the expected PH of PHPK munitions.
- The target posture, defilade state, and movement affect the PH of PHPK munitions.
- The shooter movement affects the expected PH of PHPK munitions.
- The distance between the shooter and the target affects the expected PH of PHPK munitions.
- The shooter posture does not affect the expected (PH of PHPK munitions.

Again, all of these results were consistent with the model's intended behavior.

c. LOF of Planned Direct Fire at Target

In the case of "Planned Direct Fire at a Target" missions, the user, rather than the model, chooses the target and characteristics (weapon, timing, etc.) of the engagement. Planned Direct Fire at a Target missions cannot be planned until the shooter has detected the target. After this detection, the mission and its interaction with LOF will work the same way as Auto Direct Fire.

The LOF of Planned Direct Fire at a Target algorithm was tested, as shown in Table 4, using soldiers with M16 rifles firing at enemy soldiers under Planned Direct Fire missions.

JCATS outputs were evaluated based on the following measures:

- Whether the target was successfully acquired by the shooter,
- The expected PH value when the shooter fired his weapon, and
- The effects, if any, of the munition on the target.

The tests and the results were the same as for Auto Direct Fire discussed previously. One exception occurred in version 2.3, where a problem arose when the shooter was in pop-up, but that problem has been fixed and the tests in this group all passed.

Table 4. LOF of Planned Direct Fire at Target Vignettes

Vignette ID	Vignette Test	Setup Notes
VV15	target posture: seer standing, target in 7 postures	Use M16 rifle. Seven pairs of seer/target separated by thin buildings.
VV16	target posture: seer crouching, target in 7 postures	Use M16 rifle. Seven pairs of seer/target separated by thin buildings.
VV17	target posture: seer in pop-up, target in 7 postures	Use M16 rifle. Seven pairs of seer/target separated by thin buildings.
VV18	defilade: seer standing, target in partial or full defilade	Use M16 rifle. Two pairs of seer/target separated by thin buildings.
VV19	full blockage: seer & target standing, 6 types of blockage	Use M16 rifle. Six pairs of seer/target separated by thin buildings.
VV20	multiple blockage: seer & target standing, 5 types of blockage in 6 combinations	Use M16 rifle. Five pairs of seer/target separated by thin buildings.
VV21	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use M16 rifle. Five pairs of seer/target separated by thin buildings.
VV22	flight through window is blocked: seer & target standing, 6 types of blockage	Use M16 rifle. Six pairs of seer/target separated by thin buildings.
VV23	flight through floors & ceilings: seer & target standing, 2 cases	Use M16 rifle. Two pairs of seer/target in separate floors.

d. LOF of Planned Direct Fire at an Area

Again, "Planned Direct Fire at an Area" missions are set up and initiated by the user rather than model. However, unlike their counterpart against targets, where the target must first be detected before the user can plan the engagement, area fires can be pre-planned. The weapon fires into a user-selected area. Unlike other fire missions addressed so far, for a "Planned Direct Fire at an Area" mission the model determines whether a specific target is hit not by using the relevant expected PH values, but by actually "flying the bullet," or following is Line of Flight (LOF), through the air. If the path, of the bullet intersects an object, it hits that object. If that object is a JCATS system, then the munition's effect is determined via the appropriate PK values. If the bullet's path fails to intersect, but instead comes near enough to a system, the appropriate suppressive effects (the system stops movement, sensing, shooting, etc.) are enabled. If something other than a target (i.e., terrain features, buildings, other man-made structures) blocks the bullet's path before it intersects a system, then the engagement stops. Again, if the bullet's path neither intercepts a system or comes near enough to one, no effects are recorded. The LOF of Planned Direct Fire at an Area algorithm was tested, as shown in Table 5, using soldiers with M16 rifles firing at areas populated by enemy soldiers. In order to ensure

results against targets, we used very small firing areas that each encompassed only one enemy soldier.

Although the same weapon is employed under this mission as was used when testing the "Auto Direct Fire" and "Planned Direct Fire at a Target," algorithms, the PH value could not be used as a measure in this set of tests: the model does not use or report a PH value when employing this algorithm, as it assesses the munition's effect by "flying the bullet" rather than using the relevant PH value. Instead, JCATS outputs were evaluated based on the following measures:

- Whether the weapon was fired,
- Whether the shot was blocked, and
- The effects, if any, of the munitions on the targets in the area.

Table 5. LOF of Planned Direct Fire at Area Vignettes

Vignette ID	Vignette Test	Setup Notes
VV24	target posture: seer standing, target in 7 postures	Use M16 automatic rifle. Seven pairs of seer/target separated by thin buildings.
VV25	target posture: seer crouching, target in 7 postures	Use M16 automatic rifle. Seven pairs of seer/target separated by thin buildings.
VV26	target posture: seer in pop-up, target in 7 postures	Use M16 automatic rifle. Seven pairs of seer/target separated by thin buildings.
VV27	defilade: seer standing, target in partial or full defilade	Use M16 automatic rifle. Two pairs of seer/target separated by thin buildings.
VV28	full blockage: seer & target standing, 6 types of blockage	Use M16 automatic rifle. Six pairs of seer/target separated by thin buildings.
VV29	multiple blockage: seer & target standing, 5 types of blockage in 6 combinations	Use M16 automatic rifle. Five pairs of seer/target separated by thin buildings.
VV30	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use M16 automatic rifle. Five pairs of seer/target separated by thin buildings.
VV31	flight through window is blocked: seer & target standing, 6 types of blockage	Use M16 automatic rifle. Six pairs of seer/target separated by thin buildings.
VV32	fight through floors & ceilings: seer & target standing, 2 cases	Use M16 automatic rifle. Two pairs of seer/target in separate buildings.

While testing version 2.3, we encountered a problem when the shooter was in pop-up. That problem was corrected in version 3.0, and all the tests in this group passed. The following results were recorded while testing the LOF (of planned Direct Fire at an Area) algorithm:

- No LOS is required for this mission. The weapon is fired at an area based on its coordinates.
- The algorithm's output is indifferent to target posture, movement, or defilade.
- The algorithm's output is indifferent to shooter posture (including pop-up) or movement.
- LOF can be blocked by terrain, fences, buildings, exterior walls and doors.
- LOF is not blocked by the first and second interior walls or doors of a building, but is blocked by the third interior wall or door.
- LOF is blocked by floors and ceilings (and such a mission can be planned and tested)
- LOF can go through windows into a building. Whether a potential target as seen through a window has his head above the window or not has no effect on this mission.
- Vegetation does not block LOF, no matter how dense and no matter what its PLOSB value.

Again, the results for these tests show that Planned Direct Fire at an Area works as expected.

e. LOF of Planned Indirect Fire

Planned Indirect Fire missions are performed using artillery-type munitions. To plan an indirect fire mission, the user specifies the number and type of rounds to be fired and lays down an impact line along which he wishes the rounds to detonate. The model then fires the rounds uniformly along the impact line. The actual location at which any single round lands, however, is determined stochastically, based on the specific characteristics of the weapon and munition. The round's LOF is checked during the course of its flight to determine whether its path is blocked by terrain or man-made structures. If the path is blocked, the engagement ends. Once the munition detonates, the model calculates the effects of the explosion within an area determined by the round's specific characteristics. Entities within that area, in turn, may be affected (killed or suppressed) by the munition.

Originally, the LOF of Planned Indirect Fire algorithm was tested using soldiers with a M79 grenade launcher firing at target lines near enemy soldiers. However, we had difficulties using the grenade in testing blockage of LOF, so for those tests we used the MLRS indirect fire system, employing a rocket as the munition. This provided a flat trajectory and made it easier to block the LOF. Table 6 presents vignettes used in this series of tests.

In order to get results against specific targets we used very small impact lines, each of which would affect only one enemy soldier at a time. The MLRS rocket is an area munition, and therefore does not employ or produce PH values.[14] Instead, as in the previous set of tests, JCATS outputs were evaluated based on the following measures:

- Whether the weapon was fired,
- Whether the shot was blocked, and
- The effects, if any, of the munition on the targets in the area.

Table 6. LOF of Planned Indirect Fire Vignettes

Vignette ID	Vignette Test	Setup Notes
VV33	target posture: seer standing, target in 7 postures	Use M79 grenade launcher automatic rifle. Seven pairs of seer/target separated by thin buildings.
VV34	defilade: seer standing, target in partial or full defilade	Use MLRS. Two pairs of seer/target separated by thin buildings.
VV35	full blockage: seer & target standing, 6 types of blockage	Use MLRS. Six pairs of seer/target separated by thin buildings.
VV36	multiple blockage: seer & target standing, 5 types of blockage in 6 combinations	Use MLRS. Five pairs of seer/target separated by thin buildings.
VV37	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use M79 grenade launcher automatic rifle. Five pairs of seer/target separated by thin buildings.
VV38	flight through window is blocked: seer & target standing, 6 types of blockage	Use MLRS. Six pairs of seer/target separated by thin buildings.
VV39	fight through floors & ceilings: seer & target standing, 2 cases	Use M79 grenade launcher automatic rifle. Two pairs of seer/target in separate buildings.

Most of the results in this group of tests were similar to those found in the previous section. The following results were found while testing the LOF (of Planned Indirect Fire) algorithm and are in accord with the model's expected behavior:

- No LOS is required for this mission. The weapon is fired at a target line based on its coordinates.
- The algorithm's output is indifferent to the effect of target posture, movement, or defilade.
- The algorithm's output is indifferent to the effect of shooter posture (including pop-up) or movement.
- LOF can be blocked by terrain and fences
- LOF can be blocked by buildings.

[14] See Appendix C.

- LOF can go through windows into a building. Whether a potential target as seen through a window has his head above the window or not has no effect on this mission.
- Vegetation does not block LOF, no matter how dense and no matter what its PLOSB value.

While the overwhelming majority (over 80 percent) of the tests in this group passed, a few problems were encountered during the testing of the LOF of Planned Indirect Fire algorithm:

- When the target line was drawn over top of a building, we could not carry out the mission and received a "mission aborted, target out of range" error message.[15] This problem prevented the testing of two of the vignette tests.
- We tried testing the blocking of LOF of a grenade, but were unable to set up the test. Grenades are handled differently than other munitions in JCATS, but the documentation did not present us with a full description of those differences.

f. LOF of Direct Support with Forward Observer (FO)

Direct Support with Forward Observer (FO) missions are very similar to automatic Indirect Fire missions. The key difference lies in the crucial role played here by the FO in the engagement process. Specifically, the FO must have LOS to the target before the engagement can begin. For instance, as was the case earlier with the LOS algorithm, if the FO cannot see the head of a target, it will not acquire the target and, therefore, will not call for Direct Support Fire. Once a target is acquired, the FO first looks inside his own task force for a system that can provide Direct Support and then in other task forces on his side. After identifying a suitable Direct Support system, the FO relays the coordinates of the target to this system, which then fires at the specified aimpoint coordinates. Again, as with the automatic Indirect Fire mission, the actual location at which the round lands is the result of a stochastic calculation based on the characteristics of the specific Direct Support system. The LOF for this mission type works the same way as it does for the LOF of Planned Indirect Fire mission.

The LOF of Direct Support mission with FO algorithm was tested in an identical manner to the Planned Indirect Fire algorithm (i.e., using soldiers firing M79 grenade launchers or employing an MLRS launched as a rocket firing at impact lines near enemy soldiers). Table 7 presents the vignettes used in this series of tests.

[15] The model does not allow the execution of an indirect fire mission originating from outside a building and designed to land inside the building. However, one can plan an indirect fire mission designed to land on the roof of a building.

JCATS outputs were evaluated based on the following measures:

- Whether the weapon was fired,
- Whether the shot was blocked, and
- The effects, if any, of the munition on the targets in the area.

Table 7. LOF of Direct Support with Forward Observer (FO) Vignettes

Vignette ID	Vignette Test	Setup Notes
VV40	target posture: seer standing, target in 7 postures	Use 120 mm with regular ammo automatic rifle. Seven pairs of seer/target separated by thin buildings.
VV41	defilade: seer standing, target in partial or full defilade	Use 120 mm with regular ammo automatic rifle. Two pairs of seer/target separated by thin buildings.
VV42	full blockage: seer & target standing, 6 types of blockage	Use MLRS. Six pairs of seer/target separated by thin buildings.
VV43	multiple blockage: seer & target standing, 5 types of blockage in 6 combinations	Use MLRS. Six pairs of seer/target separated by thin buildings.
VV44	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use 120 mm with regular ammo automatic rifle. Five pairs of seer/target separated by thin buildings.
VV45	flight through window is blocked: seer & target standing, 6 types of blockage	Use MLRS. Six pairs of seer/target separated by thin buildings.
VV46	flight through floors & ceilings: seer & target standing, 2 cases	Use 120 mm with regular ammo automatic rifle. Two pairs of seer/target in separate buildings.

Overall, nearly 80 percent of the tests were successful in this group. The problems found in testing this algorithm were also identical to those found for Planned Indirect Fire algorithm, and two of the vignette tests were prohibited in this group as well. See the previous section for a discussion of problems experienced with both of these algorithm categories.

g. LOF of Direct Fire with Laser Designator (LD)

Direct Support with Laser Designator (LD) missions are treated by the model in a manner identical to Direct Fire missions. Once again, the LD must first have an LOS to the target before the engagement can begin. Once the LOS is established, the LD acquires the target, calls for fire from any Direct Support-capable system in its task force,[16] and

[16] Unlike the Direct Support with FO mission, the LD does not look outside its own task force for a Direct Support system.

then shines the laser on the target. Again, the laser must also have unblocked LOS to the target. The Direct Support-capable system then fires a round which "rides" the laser beam onto the target. The round/munition effects are modeled as a PHPK munition. The LOF for this mission works the same way as it does for the LOF of Planned Direct Fire at a Target.

The LOF of Direct Support mission with LD algorithm was tested using soldiers as Laser Designators and 120 mm mortars as the associated shooters firing Precision-Guided copperhead munitions. Table 8 presents vignettes used in this series of tests.

Table 8. LOF of Direct Fire with Laser Designator (LD) Vignettes

Vignette ID	Vignette Test	Setup Notes
VV47	target posture: seer standing, target in 7 postures	Use 120 mm with regular ammo automatic rifle. Seven pairs of seer/target separated by thin buildings.
VV48	defilade: seer standing, target in partial or full defilade	Use 120 mm with regular ammo automatic rifle. Two pairs of seer/target separated by thin buildings.
VV49	full blockage: seer & target standing, 6 types of blockage	Use 120 mm with regular ammo automatic rifle. Six pairs of seer/target separated by thin buildings.
VV50	multiple blockage: seer & target standing, 5 types of blockage in 6 combinations	Use 120 mm with regular ammo automatic rifle. Five pairs of seer/target separated by thin buildings.
VV51	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use 120 mm with regular ammo automatic rifle. Five pairs of seer/target separated by thin buildings.
VV52	flight through window is blocked: seer & target standing, 6 types of blockage	Use 120 mm with regular ammo automatic rifle. Six pairs of seer/target separated by thin buildings.
VV53	flight through floors & ceilings: seer & target standing, 2 cases	Use 120 mm with regular ammo automatic rifle. Two pairs of seer/target in separate buildings.

Although this is a PHPK munition, the value of the PH is not reported in the output file, and therefore we were unable to use PH as a measure of the results. JCATS outputs were evaluated based on the following measures:

- Whether the weapon was fired,
- Whether the shot was blocked, and
- The effects, if any, of the munition on the target.

Although fewer than half the tests passed in this group, we were able to show that the target posture, defilade state, and movement do not affect the results. This turned out to be the worst performing group; however, we determined that all the problems were

associated with how this fire mission works with buildings. Specifically, the model fails to check line of flight for laser-designated missiles, allowing them to attack targets directly behind a building as well as targets inside a building. In both cases, the line of flight might be blocked, although the line of sight from the laser designator to the target may not be blocked. All nineteen of the tests that failed in this group entailed some element of this problem.

h. Soldier Movement

This group of tests encompassed a variety of soldier movement algorithms and subroutines, including movement over various types of terrain, through rubble, along ramps and inside buildings. This group also examined algorithms involving breaching and penetrating engineering objects (e.g., wire, other obstacles).[17] Table 9 presents Soldier Movement Vignettes.

Table 9. Soldier Movement Vignettes

Vignette ID	Vignette Test	Setup Notes
VV54	posture & terrain with no micro terrain; 3 postures, 3 terrain inclines	Use 9 soldiers
VV55	posture & terrain on road; 3 postures, 3 terrain inclines	Use 9 soldiers
VV56	posture & terrain on grass; 3 postures, 3 terrain inclines	Use 9 soldiers
VV57	posture & terrain on "other" terrain (woods, shallow water, waste-deep water); 3 postures, 3 types of vegetation, flat ground	Use 14 soldiers
VV58	blocking, breaching & penetration: soldier walking on road, 6 types of blockers in 14 cases	Use 14 soldiers. Use engineering object with Breach code (B)=0 Penetrate code (P)=0 for no B or P, otherwise use object with both B and P capability and turn breach on to breach and off to penetrate.
VV59	movement in buildings: soldier walking inside building, 2 blocking entities, breach and penetrate	Use 6 soldiers.
VV60	entering building: (1) 3 postures, soldier enters through exterior window, (2) 2 postures, soldier entering via one story ramp, (3) 3 postures, soldier entering via 3 story ramp	Use 8 soldiers.
VV61	rubble: 3 postures, 3 terrain inclines.	Use 9 soldiers.

[17] Breaching an obstacle provides an opening in the obstacle for follow-on forces to exploit and move through. Penetration does not provide a permanent opening; i.e., follow-on forces must also penetrate the obstacles.

Out of the 74 different tests conducted under this group, 73 were judged to be successful. The following results were recorded while testing the algorithms:

- Soldier speed reflects the level selected by the user; i.e., slow, moderate, or fast.
- Soldier speed is affected by the steepness of the terrain.
- Soldier speed is affected by the micro terrain, such as roads, grass and woods.
- Soldier speed is affected by the water.
- A soldier will breach if he has the capability and the Breach Option is turned on.
- A soldier will penetrate if he has the capability and the Breach Option is turned off or if he does not have breach capability.
- Vegetation does not block the movement of a soldier.
- Vehicles do not stop the movement of people and vice versa. They move around each other.
- A soldier can move within buildings, both on a floor and from one floor to another.
- A soldier can enter a building through a window.
- A soldier can move up a ramp and his speed may decrease based on the incline of the ramp.
- Soldier speed is affected when moving through building rubble.

The one significant problem found in version 3.0 while testing this algorithm involved an aspect of the breaching operation. Specifically, a follow-on entity/system is allowed a free pass through a breach in progress. In other words, the follow-on system may pass immediately through the breach even though the first system has yet to complete the breaching operation.

i. Vehicle Blocking

The tests in this group were designed to test whether vehicles could block the movement of other vehicles when the 'vehicle block movement' parameter is set to 'on'. Table 10 describes the vignettes used in this series of tests.

Due to computational burdens placed on the model by this activity, the user must specifically toggle it on during the course of the simulation, otherwise vehicle blockage does not occur.

Table 10. Vehicle Blocking Vignettes

Vignette ID	Vignette Test	Setup Notes
VV62	blocking in column: 3 tanks in 6 cases of movement	Use 6 columns of 3 tanks each
VV63	Other Blocking: 2 tanks, 2 soldiers, or a tank and a soldier	8 cases of blocking

All tests in this series passed, as the model performed as expected. The following results were recorded while testing the blocking algorithm:

- A vehicle that overtakes another vehicle on the same path will be blocked from moving until the overtaken vehicle moves.
- Two vehicles that meet each other will block each other until one moves away. Such vehicles are able to move away from each other at a 90-degree angle.
- Vehicles do not stop the movement of people and vice versa. They move around each other.
- A stationary vehicle will block another vehicle whose path crosses it.

j. Miscellaneous

The tests in this group included various characteristics of ramps, fences, and stairs. The movement of soldiers over stacked terrain polygons was also examined. The tests, described in Table 11, were as follows:

- Testing of ramps, fences, and stairs, and
- Testing of movement of soldiers over stacked terrain polygons.

Table 11. Miscellaneous Vignettes

Vignette ID	Vignette Test
VV65	Bullet proof workaround option 2
VV66	Test LOS and LOF on ramp
VV67	Test LOS and LOF on “stairs”
VV68	Test to see if can penetrate fence and if type of material used for fence matters.
VV69	Test stacking of terrain
VV70	Test LOF on fences with different materials.

Again, all tests in this group passed, as the modeled performed as expected. The following results were recorded:

- A soldier on a ramp can be acquired and fired upon.

- Soldiers on stairways cannot be visualized on the JCATS screen because no such physical element exists in the model. Instead, movement on stairs/between floors is modeled by time delays on a Go To Floor movement node.
- The type of material used for a fence does not directly affect breaching or penetration but each type of fence can be assigned a different terrain code, which in turn determines the time to breach or penetrate the fence.
- There was a change in speed as a soldier moves from one set of terrain polygons to another.
- The material of a fence does not affect whether it can block LOF.

Finally, we examined a pair of proposed workarounds for modeling bulletproof glass. We discovered that neither of these two workarounds addressed Direct Fire missions, although both worked for Indirect Fire engagements. The first option was to create an external see-through door. This option worked for Indirect Fire missions because indirect fire rounds will be stopped by any door (blocks LOF) regardless of its composition/visibility. It fails to work for Direct Fire missions, however, as LOS is achieved, which in turn implies LOF, and the bullet penetrates the glass. The second option involved the creation of three see-through internal walls behind a window. Again, the Indirect Fire rounds combined with see-through walls will work, as the round will always be stopped by the third interior wall. But, again, it fails with Direct Fire missions: If the three walls are see-through, we get LOS and automatically get LOF and the bullet passes through the glass. We did not consider that the vignettes testing the bulletproof glass workarounds failed, as they were intended simply to examine proposed workarounds.

2. Validation

The validation process should assess whether JCATS provides a sufficient approximation to the real world. Our validation effort, described earlier, was conducted by a group of Subject Matter Experts at the Constructive Simulation Center, Dismounted Battlespace Battlelab (DBBL) in Ft. Benning, Georgia. In developing the scenarios to be employed in this effort, DBBL chose to use the Objective Force Warrior (OFW) Situational Awareness (SA) force structure. This force structure includes some advanced technologies not currently fielded by the Army. These technologies included through-wall sensors, Kinetic Energy (KE) munitions, robotics, PGMM, and Directed Energy munitions. Fortunately, DBBL had just recently conducted sixty production runs in support of the OFW-SA study. Furthermore, because of the work already performed under this study, DBBL had two different types of missions to examine: an attack

scenario and a defensive scenario. The two scenarios were selected because they best represented the six scenarios identified by IDA for the MOUT validation.

Twenty-two Subject Matter Experts (SME's) were identified. Twenty-one of these were infantrymen with an average of sixteen years of military service. The SME's were shown the replays of the selected simulation production runs and excerpts from the "datevent" files[18]. They were given access to a JCATS client workstation and a qualified operator, and were permitted to "use" JCATS. Each SME was asked to complete one or both parts of a two-part validation questionnaire. [1] . The first part of the questionnaire—addressing operational validation—was designed to assess whether the JCATS output sufficiently represent the "real" world of urban combat. This effort was accomplished by employing subject matter experts (SME's) with knowledge of, and familiarity with urban operations. The second part of the questionnaire—addressing structural validation—was designed to assess whether the model's code, editors, and post-processing capabilities were adequate for representing the "real" world of urban combat (e.g., is the terrain resolution adequate for modeling urban operations). Many of the aforementioned SME's also had considerable experience conducting and observing JCATS gaming and had in-depth understanding of the JCATS code. These SME's were requested to complete the structural validation portion of the questionnaire as well. Each question was to be answered on a one to five scale, with one meaning "not at all" and five meaning "very well."

The results, averaged over the respondents, were as follows:

[18] "Datevent" files are log files generated by JCATS while a scenario is being run. The file captures events that the user has selected to appear in that file, such as LOS, shots fired, hits, and kills.

[1] The Joint Non-Lethal Weapons Directorate conducted a V&V that was completed in October of 2000. The questions and statements used for the validation portion of that effort were used as a starting point for these questionnaires.

	OPERATIONAL VALIDATION	AVERAGE SCORE
1	**Does JCATS produce results that are feasible?**	4.68
2	**Does a difference in the input produce the expected proportional change in the output?**	4.59
3	**Do the levels of force structure and interaction have sufficient fidelity and resolution?**	4.59
4	**Based on your military experience, does JCATS compare favorably to historical, test, laboratory, and/or exercise data?**	4.45
5	**Does JCATS adequately represent a MOUT environment?**	4.55
6	**Is JCATS suitable for the overall intended use as an analytical tool?**	4.86

	STRUCTURAL VALIDATION	AVERAGE SCORE
1	**Is JCATS sensitive to the data input values?**	4.58
2	**Does JCATS adequately represent the real world?**	4.42
3	**Is JCATS complete and are the functions adequately modeled?**	4.37
4	**Is there adequate and consistent representation of terrain and environment across all JCATS components?**	4.37
5	**Can JCATS output/results be used clearly, adequately and appropriately to address MOUT problems?**	4.72
6	**Can JCATS runs be accomplished and results analyzed in a timely manner?**	4.63
7	**Are baseline scenarios, terrain data, threat data, and weapon performance data for JCATS database available?**	4.47
8	**Are terrain and environment representations functionally adequate to address MOUT issues?**	4.53
9	**Are the clarity, fidelity, complexity and level of detail of the simulated entities acceptable for its intended usage?**	4.56

These results suggest that the SME's strongly endorsed the view that both the operationally and structurally the JCATS model passed the validation test. In other words, the results suggest that the SME's felt that the representation of urban combat

found in JCATS sufficiently and adequately represented the "real" world of MOUT operations.

D. Conclusions

Overall, we conclude that JCATS MOUT-related representations successfully passed both the verification and the validation examinations. The verification results strongly suggest that JCATS successfully demonstrated that its MOUT-associated algorithms performed as expected. Out of a total of 424 tests, we determined that over 93 percent of the results were consistent with the intended behavior of the model. The majority of the failures (19 out of 29) occurred during testing of a single algorithm, and were concerned, in whole or in part, with the fact that the model fails to check line of flight for laser designated missiles. The remaining failures were of minor consequence, and none of the failures was judged to be "fatal" in the sense that they caused the simulation to "crash," they performed a calculation incorrectly, or that they pertained to any but a very specialized operation, task, or function. These errors have been reported to LLNL and will be addressed in later versions of the model. Likewise, the validation results strongly suggest that the model adequately represents the realities of combat in an urban environment. The survey results from a group of urban combat SME's strongly endorsed the use of the model for these types of operations.

Again, this V&V of the JCATS model, combined with other efforts (e.g., the Non-Lethal Weapons JPO V&V), provides the basis for judgment on the part of managers and users with respect to acceptance or accreditation for a specific intended purpose: i.e., analyses addressing MOUT operations. Having presented the evidence, we will leave it to the relevant individuals to determine whether the model can be accredit for their particular study.

As with any large, constantly evolving model, the V&V of JCATS is a on-going process; not every element of the model has yet been reviewed (e.g., littoral warfare) and changes or additions to the model occur regularly. On the latter point, one caveat should be noted: Since this V&V was completed a minor modification to the model has been released (version 3.1). The latest version of JCATS (version 4.0) was released in October 2002. Major changes in 4.0 include a new detection model (ACQUIRE) and the addition of nuclear weapons. Few, if any, changes, however, were made to the specific algorithms examined in this V&V study. Future users of the model, nonetheless, may want to check any minor modifications made to these algorithms in a MOUT context, as well as the

major changes and additions made to other algorithms in 4.0, prior to their accreditation of JCATS for analyses entailing urban operations.

GLOSSARY

GLOSSARY

ACTD	Advanced Concept Technology Demonstration
AOF	Angle of Flight
ASAP	As Soon As Possible
BOI	Basis of Issue
C4ISR	Command, Control Communications, Computers, Intelligence, Surveillance and Reconnaissance
CEP	Circular Error Probable
DDBL	Dismounted Battlespace Battle Laboratory
DF	Direct Fire
DIS	Distributed Interactive Simulation
DS	Direct Support
DSB	Defense Science Board
DTED	Digital Terrain Elevation Data
FER	Force Exchange Ratio
FFA	Free Fire Areas
FO	Forward Observer
FP	Firepower
HE	High Explosive
HLA	Higher Level Architecture
ICM	Improved Conventional Munition
IF	Indirect Fire
JCATS	Joint Conflict and Tactical Simulation
JPO	Joint Program Office
JRTC	Joint Readiness Training Center
JTS	Joint Tactical Simulation
KE	Kinetic Energy
KK	Critical Kill
LD	Laser Designator
LER	Loss Exchange Ratio
LLNL	Lawrence Livermore National Laboratory
LOF	Line of Flight
LOS	Line of Sight
MAF	Mobility and Firepower
MOB	Mobility
MLRS	Multiple Launch Rocket System
MOUT	Military Operations in Urban Terrain
NFA	No Fire Areas
NVEOL	Night Vision and Electro-Optics Laboratory
OFW	Objective Force Warrior
OOTW	Operations Other Than War
PD	Probability of Detection
PDF	Planned Direct Fire
PGMM	Precision Guided Mortar Munition

PH	Probability of Hit
PHPK	Probability of Hit, Probability of Kill
PIF	Planned Indirect Fire to Area
PK	Probability of Kill
PLOSB	Probability of Line of Sight Blockage
POW	Prisoner of War
RLEM	Rifle-Launched Entry Munition
ROE	Rules of Engagement
SA	Situational Awareness
SME	Subject Matter Experts
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
V&V	Verification and Validation

APPENDIX A

JCATS V&V PLAN FOR MOUT

Institute for Defense Analyses

A. Introduction

The JCATS verification and validation (V&V) effort conducted for the MOUT ACTD will be two-fold. The first step will be to oversee piecemeal V&V efforts currently under way; the second step will be to pull together current and past JCATS-related V&V work, and identify gaps in the coverage of MOUT-related capabilities. Once identified, the team will undertake efforts to fill in these gaps through the methodology described below. Briefly, the verification portion of this methodology entails a check of the JCATS code and the use of tightly focused vignettes; while the validation portion involves a mix of insights and observations from subject matter experts (SMEs) intimately familiar with urban combat, and data collected from training and field experiments. Understanding the difficulties involved in a full-fledged V&V of a force-on-force model, the goal of this effort will be to determine the reasonableness of JCATS for MOUT representation. Can a group of MOUT SMEs reach a consensus that the model adequately represents combat in an urban environment? Before describing the proposed approach in more detail, we will examine why this approach is the best available for conducting a V&V of JCATS.

It is important to understand, that V&V activities provide the basis for judgment on the part of managers and users with respect to acceptance or accreditation for an intended purpose. The V&V record provides the basis for that judgment.

B. Approach

As pointed out in the recent Defense Science Board (DSB) study on M&S[1], a range of M&S types exist, covering a variety of items and levels, from detailed engineering-level models of a single combat system or subsystem up through force-on-force models examining combat on the scale of a joint task force or larger. JCATS falls somewhere in between these extremes, being approximately (following the phrasing of the report) a conventional small unit (up to brigade) type model. The report argued that the different types of models serve different purposes and suggested that, therefore, a range of corresponding V&V criteria or approaches should be adopted. Specifically, the report called for "more adaptability in V&V approaches as described in DoD VV&A instructions." We agree with this general approach and have adopted it for the JCATS V&V effort.

[1] Defense Science Board, *Report of the Defense Science Board Task Force on Advanced Modeling and Simulation for Analyzing Combat Concepts in the 21st Century*, OSDA&T, May 1999.

Verification

The Verification, Validation and Accreditation of Army Models and Simulations Pamphlet (DA PAM 5-11) defines verification as the process thereby establishing that the M&S code and logic correctly perform the intended functions[2]. To confirm that the model is performing as expected (i.e., the verification portion of the V&V), we plan to provide both logical and code verification. The code verification effort will include an algorithm check and a peer review of the code, as they are defined in DA PAM 5-11. However, a simple check of the coding for the algorithms will be sufficient only in certain cases. In others, owing to modern programming techniques, an intimate knowledge of the broader programming environment (e.g., ways in which data are being cached within the model, etc.) is required to adequately verify that an algorithm is performing as expected. Unfortunately, time and resources do not allow for the development of such expertise in this V&V effort. An alternative approach, to be utilized here where a simple check of the algorithm's implementation in the code is insufficient, is to examine the model's output data resulting from the performance of a single algorithm or function to ensure that the model is performing that function as expected. In pursuit of this approach, we will construct and run a set of tightly focused vignettes in JCATS, each designed to examine a single JCATS function or algorithm.

Validation

The DSB M&S study focused on the need to adapt the appropriate validation criteria or approach to each type of model. For force-on-force models such as JCATS, the study suggested a validation approach that combined some mix of a pure "analytical comparison to the known" (the traditional V&V approach) and "operational judgment about reasonableness." Again, we have adopted this approach in our validation efforts.

Indeed, for the lowest item level models, engineering-level representations of systems or their components (e.g., a tank main gun or a radio), the comparison of model interactions and outputs with actual real world performance data derived from the systems being modeled is the most appropriate and effective method for validating such a model (even though it may be costly and time-consuming), as the DSB study suggests. In these cases, the items under study follow the laws of physics and engineering; the confounding effects of human interactions can be safely ignored. Constructing and executing tightly controlled, realistic, and replicable experiments on the actual systems

---

[2] Headquarters Department of the Army, *Verification Validation, and Accreditation of Army Models and Simulations*, Pamphlet 5-11, October 1993.

being modeled is comparatively easy, as is the collection of detailed performance data for comparison with the model outputs. With such models, SME judgments and observations play little role. In fact, extensive use of such observations would be inefficient and redundant at best, misleading at worst.

The situation is quite different, however, for force-on-force models such as JCATS. First, "live" experimentation at this level is often artificial and can rarely replicate the full demands of real combat, being limited by environmental, safety and other concerns. As the DSB study comments, such experiments are often little more than simulations themselves. Live experimentation at this level also can be very costly in terms of time, money, and troops. Scheduling large numbers of soldiers, for example, for extensive periods of experimentation is extremely difficult. Moreover, as a recent GAO report[3] concluded, the Armed Services have few MOUT training and experimentation sites, and all have limited urban representation (the largest is village size and contains buildings no taller than a few stories in height). And, much of the data required for a complete JCATS V&V would still need to be generated: an extensive library of data on urban operations does not exist, as the Services have only begun within the last few years to conduct experiments and training in MOUT environments. However, experiments held at these facilities can provide insights useful to a portion of the JCATS V&V problem, and the results of previous and on-going experimentation at these sites can aid in this effort. But, to reiterate, the limitations and costs of such experiments mean that they cannot be the full, or even the major, answer to the JCATS V&V effort.

As an alternative to live experiments, history might provide a realistic laboratory for comparing model outputs. However, to represent an historical battle in a force-on-force model to the level necessary for activities such as V&V requires a tremendous amount of detail concerning the engagement. In only a very few cases have such detailed data been collected. The Persian Gulf War's "Battle of 73-Easting," a mechanized/ armored engagement that occurred in the deserts of Iraq, is probably the best documented example. A data collection team surveyed the battlefield within days after the engagement occurred, before the dead hulks, spent TOW lines, and vehicle tracks could be removed from the sand. Data derived from navigation equipment on board U.S. vehicles were able to precisely record their location as the battle took place. U.S. participants were extensively interviewed immediately after the war on the details of the

[3] United States General Accounting Office, *Military Capabilities: Focused Attention Needed to Prepare U.S. Forces for Combat in Urban Areas*, GAO/NSIAD-00-63NI, February 2000.

engagement and were asked to comment several times over the next year on the resulting scenario/narrative as it was being developed. As a result, modelers had precise information on the locations of all participating vehicles throughout the course of the battle, knew generally where Iraqi dismounted troops were positioned, possessed detailed knowledge of the timing and number of rounds fired by U.S. forces as well as general information on Iraqi anti-armor rounds fired, and had data for both sides on vehicle hits and kills. This description is provided to illustrate the level of detail required to accurately recreate historical engagements in models such as JCATS.

Unfortunately, no urban battles have been so closely studied and documented. Given the rapid, close-quarter, individual-combatant nature of warfare in urban environments, the development of such a detailed account of an urban engagement, in fact, is probably an impossible task. And, in the absence of such detailed data, models can often be "tweaked" to provide the general outlines of the historical outcome; a result which says little about the verisimilitude of the model. Finally, both live experimentation and historical cases (were they to be found) provide only a limited number of urban combat scenarios and conditions in which to test the model; they beg the question as to whether a change in conditions, including a change in the decisions made by humans on the ground, would cause the model to "fail."

A third and final source for comparing force-on-force models like JCATS with "real" combat is to garner insights and rely on the judgments of subject matter experts with intimate knowledge of, and familiarity with, the type of combat under study. For MOUT operations, these experts would include individuals with considerable experience conducting and observing urban training exercises, such as the personnel at the Joint Readiness Training Center (JRTC) at Fort Polk, or individuals who have been involved in actual, recent U.S. military operations in urban environments. The knowledge and experience of a select group of such people can be used to isolate and focus on key elements of urban combat. These elements can be represented in the model, and the SME's can be asked to make judgments both on the operations as they take place on the JCATS screen as well as on the model's processed output. The advantage to this approach is that a wide range of urban scenarios, both large and small, can be examined relatively quickly.

The most cost-effective and sensible validation alternative for force-on-force models like JCATS, as suggested by the DSB study, is to rely on a mix of field data (collected through training and live experimentation) and observations and insights of MOUT subject matter experts. The field data provide a degree of quantitative backbone

to the essentially qualitative judgments of urban combat experts concerning the "reasonableness" of the JCATS model for representing MOUT operations. In this manner, a range of soldier tasks, small unit operations, and urban environments can be examined within practical cost and time constraints.

C. Description of V&V Approach

Our V&V approach for JCATS is grounded in a number of principles. First, we are assessing the capabilities of JCATS for MOUT operations only; other types of operations using JCATS (e.g., littoral warfare, armored maneuver warfare) will not be addressed unless they directly affect urban operations. Second, many of the JCATS algorithms are identical to those found in most of the Janus family of models. These algorithms have been used for over twenty years and are widely accepted in the modeling community. Except for coding, such algorithms need not be closely examined during our JCATS V&V effort. Finally, as mentioned earlier, a number of V&V activities have recently been or are being conducted with JCATS. These activities include an earlier V&V conducted of JCATS immediate predecessor, JTS, as well as a V&V currently being undertaken by the Army's Dismounted Battlespace Battle Laboratory (DBBL) for the Non-lethal JPO examining the capabilities of JCATS in the non-lethal arena. Many of the operations examined in these activities have applicability for MOUT and will, therefore, help contribute to our overall MOUT V&V effort. Our purpose is to provide oversight for the on-going activities, ensure that present and past V&V efforts feed into our MOUT V&V effort, identify MOUT-related areas not yet examined, undertake V&V activities in these latter areas, and then tie all of these efforts together to arrive at an overall assessment of JCATS for MOUT.

Our verification effort, as discussed earlier, will involve a mix of logical verification, algorithm code checking, peer review, and examination of tightly focused vignettes. The JCATS algorithm list has been carefully reviewed and a priority has been assigned based on MOUT. The prioritized list of algorithms is contained in Annex A. We will send a team of computer programmers to Lawrence Livermore National Laboratory (LLNL) (the JCATS developer) to examine relevant portions (algorithms) of the JCATS source code. Some of these algorithms have not been documented in the algorithm manual and will require guidance from LLNL personnel to complete the verification. LLNL personnel also will provide guidance in identifying which algorithms or functions need to be examined via the vignettes. Again, the entire model's code need not be examined in our verification effort, but only those portions which we identify as being

MOUT-related and not recently verified. A list of MOUT capabilities is provided in Annex B. These capabilities will be used to develop case studies to assess MOUT capabilities in JCATS. Once this activity is under way, the validation phase will begin as outlined above. Although the amount of field data on urban combat remains small, we have access to most of the current collection. Besides the data generated by the MOUT ACTD, we have an on-going relationship with personnel at the McKenna MOUT site at Fort Benning, who are creating the capability to continuously collect a wide variety of data on the exercises being conducted at the site. In addition, we intend to tap into their proposed project linking JCATS and the instrumented live test range at McKenna, designed ultimately to support a seamless connection between this latter site and non-lethal weapon usage in JCATS. We also intend to use data collected during the Marine Corps Urban Warrior and Project Metropolis. Finally, we have made contacts with senior personnel at the JRTC and have obtained agreement to access data collected during unit rotations through this training facility. All of these data will be used to calibrate and check portions of the JCATS MOUT representation.

Again, the final assessment on the validity of JCATS for MOUT will reside in the judgment of the SMEs. The JRTC leadership has further agreed to provide SMEs for the JCATS V&V through their corps of observer controllers. Additional SME support will come from a pool of active and retired Army and Marine Corps officers with real world experience in such urban operations as Panama, Somalia, Haiti, Bosnia, and Kosovo. We envision requiring a handful of multi-day sessions with the SMEs in which we will identify the key elements of MOUT operations, obtain their insights and lessons learned from their MOUT experience, and obtain their judgments on how well JCATS represents these operations. A preliminary timeline of the V&V activities is presented below.

In conclusion, it is worth repeating that these activities will provide the record on which users and managers can rely as the basis for accepting JCATS, or not, for their specific MOUT M&S tasks. While this work may provide insights for those who want to use the model for other purposes, or who want to engage in V&V of a broader scope, we emphasize the intent and limits previously stated.

Timeline for MOUT V&V	
Algorithm Identification	June 2000
Algorithm Prioritization	June 2000
Acquire/Install JCAT 2.3	August 8-10, 2000
Planning meeting with LLNL and JNLWD	August 10, 2000
Visit LLNL, kick off logical and code verification. Request hard copy of selected algorithms.	September 24-29, 2000
Identify Subject Matter Experts (SMEs)	Late September 2000
Contact SMEs, check availability, etc.	Late September 2000
Brainstorming session -- Verification short vignettes/case studies Validation - Scenario Development and process.	October 3, 2000
Algorithm and case study verification	September 24 - December 15, 2000
Formalize Scenarios to be used for validation	October 15, 2000
IPR -- verification - (IDA MOUT team meeting)	Early November 2000
1st Validation Session	November 6-10, 2000
IPR - Validation - (Discuss/present preliminary results at IDA MOUT team meeting and make recommendations for 2nd Validation session.)	November 16, 2000
2nd Validation Session	December 4-8, 2000
Draft report of MOUT V&V (algorithms review, case study and scenario findings)	January 15, 2001

ANNEX A

JCATS ALGORITHM PRIORITIZATION

Due to limitations of time and money, we were unable to examine every algorithm in JCATS during the Verification phase of the V&V. Instead, we selected and then prioritized a set of algorithms based on the following criteria:

- Did the algorithm have relevance to an identified MOUT model capability/requirement (see Annex B)?
- Had the algorithm been examined during the Non-Lethal JPO V&V effort?
- If the algorithm had already been examined in the previous V&V effort, did the need to consider an urban environment (specifically the presence of buildings) necessitate another look at this algorithm?

The chart on the following page summarizes our prioritization effort, as well as a similar prioritization conducted by the Non-Lethal JPO for their V&V task. The chapters and algorithms listed in the far-left hand column are pulled directly out of the JCATS Algorithms Manual (as of September 1999). The manual served, in part, as guide to the expected capabilities of JCATS. However, the manual was incomplete at the time our V&V effort began; several sections needed to be written. When information about these algorithms was required, we contacted LLNL directly.

JCATS Algorithms as of 9/99

Chapter/Algorithm	Status[1]	JNLWD V&V Prioritization	MOUT V&V Prioritization[2]
CHAPTER 2 - ACQUISITION			
2.1 General		2	3B
2.2 NVEOL Optical Sensors		3/4	4B
2.3 NVEOL Thermal Sensors	TBW	5	5
2.4 Active Radar			
2.5 Active Sonar			
2.6 Passive Radar			
2.7 Passive Sonar			
2.8 Line of Sight (LOS)		1	2B
2.9 Horizon Check			
2.10 Enhanced Lighting	TBW	6	
CHAPTER 3 - ADJUDICATION OF WEAPON EFFECTS			
3.1 Point Effect (PHPK) Munitions		7-10	8B
3.2 Area Effect Munitions		11-17	9B
3.3 Environmental Effects	TBW	44	10
CHAPTER 4 - AGGREGATION			
4.1 Aggregate an Aggregate		45	
4.2 De-Aggregate an Aggregate			
4.3 Join a System Or Aggregate With an Aggregate			
4.4 Depart a System Or Aggregate From an Aggregate			
4.5 Formations			
CHAPTER 5 - CAPTURE AND SURRENDER	TBW	32/33	15
CHAPTER 6 - CASUALTY AND REPAIR	TBW	38	16
CHAPTER 7 - DEFILADE	TBW	41	12
CHAPTER 8 - ENVIRONMENT	TBW		1
8.1 Barriers and Minefields	TBW	34/35	
8.2 Light	TBW	36	
8.3 Weather	TBW	37	
CHAPTER 9 - FATIGUE	TBW	40	
CHAPTER 10 - FRATRICIDE	TBW	39	
CHAPTER 11 - MOUNT			
11.1 Mount Passenger			
11.2 Mount Crew			
11.3 Dismount Passenger or Crew			
11.4 Dismount All Passengers or Crew			
11.5 Aggregate As a Passenger			
11.6 Aggregate As a Crew			
11.7 Mounting On an Aggregate			
CHAPTER 12 - MOVEMENT			
12.1 Movement During Planning			
12.2 Movement During the Game			
12.3 Movement in the Air			
12.4 Movement on the Ground			
12.5 Movement on Water			
12.6 Movement Under Water			
12.7 Movement in Buildings			1
12.8 Stationary Systems			
12.9 Activity Nodes	TBW		11
CHAPTER 13 - POPUP	TBW		
CHAPTER 14 - SOUND	TBW		13
CHAPTER 15 - SUPPLY			
15.1 Transfer Supplies			
15.2 Re-Supply			
15.3 Level Supply			
15.4 Level Load			
15.5 Recover Ammo			
15.6 Recover Weapon		70	
15.7 Load Bomb			
CHAPTER 16 - TARGETING			
16.1 Automated Targeting		18/19	6/6B
16.2 Manual Targeting	TBW	20/21	7B

NOTES:
1. TBW = To Be Written for the Algorithms Manual

2. B after number means that the algorithm was considered during the JPO Non-Lethal V&V, however will require further consideration for MOUT (due to the presence of buildings)

APPENDIX B

NOTES ON JCATS OPERATION

Based on Meeting at
Lawrence Livermore National Laboratories
September 26-27, 2000

Institute for Defense Analyses

A. Direct Fire

There are three types of direct fire:

1. auto direct – performed automatically by the simulation
2. planned direct – by the user
3. direct support with laser designator (not discussed in this paper).

Direct support fire can be triggered by:

- forward observer and artillery (this is indirect fire; see Section B)
- laser designator and someone operating laser seeker (this is direct fire).

Note: Laser seeker can have LOS to target when it finds the target.

1. Auto Direct Fire

The steps in performing auto direct fire are as follows:

1. create a list of acquisitions
2. determine enemies from this list (if fratricide is on, use the fratricide area if at level 3 acquisition; otherwise use level 4 acquisition)
3. enumerate ways to shoot the targets
4. look at priority for engagement (high priority number is processed first). Does the engagement make sense? Do we have all elements?
5. start the engagement against an object
6. use PH mode [i.e., load the weapon with the munitions, estimate when in the future it should hit, get PH for factors (shooter, weapon, range, target & target/shooter postures: shooter moving or stationary, target moving or stationary, target exposed or in partial defilade, head shot or flank shot)]. Note 1: Other factors adjust the PH (e.g., LOS gives only partially exposed target, target defilade state affects PH as shown below). Note 2: with PH mode, the model does not 'fly the bullet' (i.e., follow the complete path of the trajectory for the munitions)
7. bullet given free pass, i.e., check that bullet does not hit any intermediate object that will stop it. There is no check on vegetation encounters such as grass
8. check LOS and then shoot
9. if the target is not, hit the target will be suppressed **if** the round lands within or passes through a specified column around the target
10. if the target is hit, go to the PK tables to determine the effect of the hit.

The PH table has two levels of defilade: exposed or defilade. JMEMS has three defilade states: exposed, partial, and full. JCATS simulation uses PH defilade to be equivalent to the JMEMS partial defilade. If the system is in full defilade then the PH is scaled to (height exposed in full defilade/height exposed in partial defilade) * PH. The defilade states are represented by the heights of the target that are exposed; full defilade gives a

lower value than partial defilade. **Note:** This algorithm has been changed in JCATS Version 3.0. See Appendix F, Problem 1, for a discussion of the new algorithm.

PK tables are based on the following:

1. munitions vs. target
2. range
3. head vs. flank
4. exposed vs. defilade
5. flavor of kill (MOB, FP, both MOB and FP, KK).

JMEMs categories:

- MOB = mobility kill
- FP = firepower kill
- KK = critical kill = dead

Note: If KK, then MOB and FP kill.

Figure 1 shows how the JCATS code uses the JMEMS data for the types of kills that are handled in the model. JCATS uses the levels of kill data provided by JMEMS to make distinct kill bins. When a kill occurs, JCATS rolls the dice to randomly decide into which bin a kill will fall. The results may be no effect (NE), firepower only (FO), mobility only (MO), mobility and firepower (MAF), or dead (D). The consistency checker will give a warning message if the input data do not fit the equations in the figure. The consistency check must be selected from the tools menu bar in order for this test to be run.

A user flag 'critical kills only' can be set in JCATS to make any kind of kill be dead. In this case, a mobility kill or a firepower kill is counted as dead.

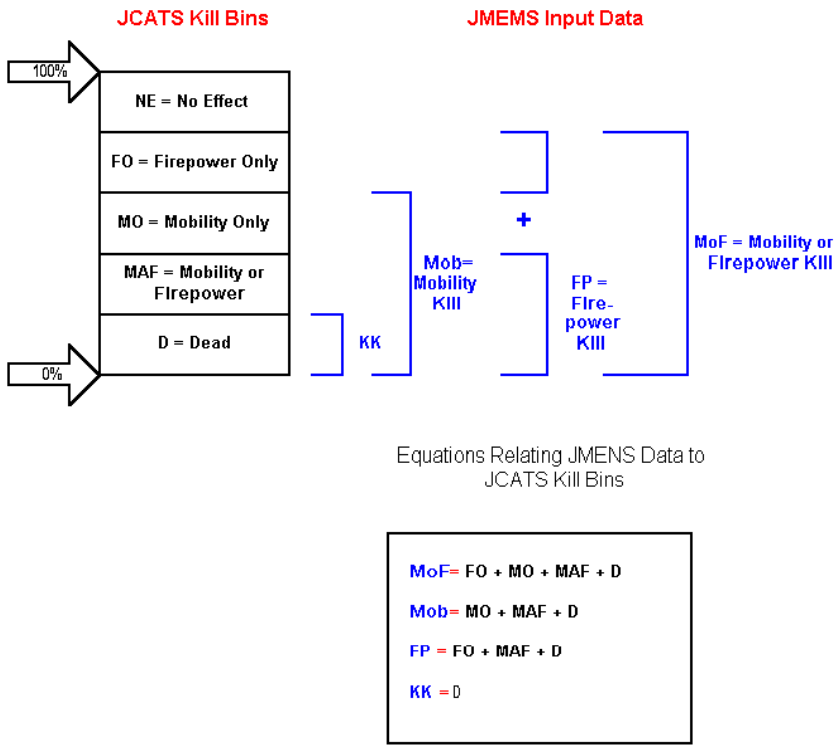

Figure 1 Relationship Between JCATS Kill Bins and JMEMS Input Data

2. Planned Direct Fire

The user can plan direct fire in two ways:

- Against a target (a 2.0+ feature)
- To an area (suppressive to keep enemy down).

a. Planned Direct Fire at a Target

Planned direct fire against a target must be considered during the simulation after the shooter has acquired the specific target. The mission then works the same way as auto direct fire. Planned direct fire against a target is not usually used for military operations but is used for police action, riot control, or a sniper against a specific target.

b. Planned Direct Fire at an Area (Suppressive Fire)

When planned direct fire goes to an area, the model 'flies the bullet' and, if it hits another target, that is the end of the flight.

Suppressive fire is defined in terms of duration of firing (e.g., 15 seconds) and rate of fire (e.g., 2 rounds per second). The weapon fires to a random point within the impact area, which is modeled as a circle (see Figure 2). Multiple ED records are created for each target hit.

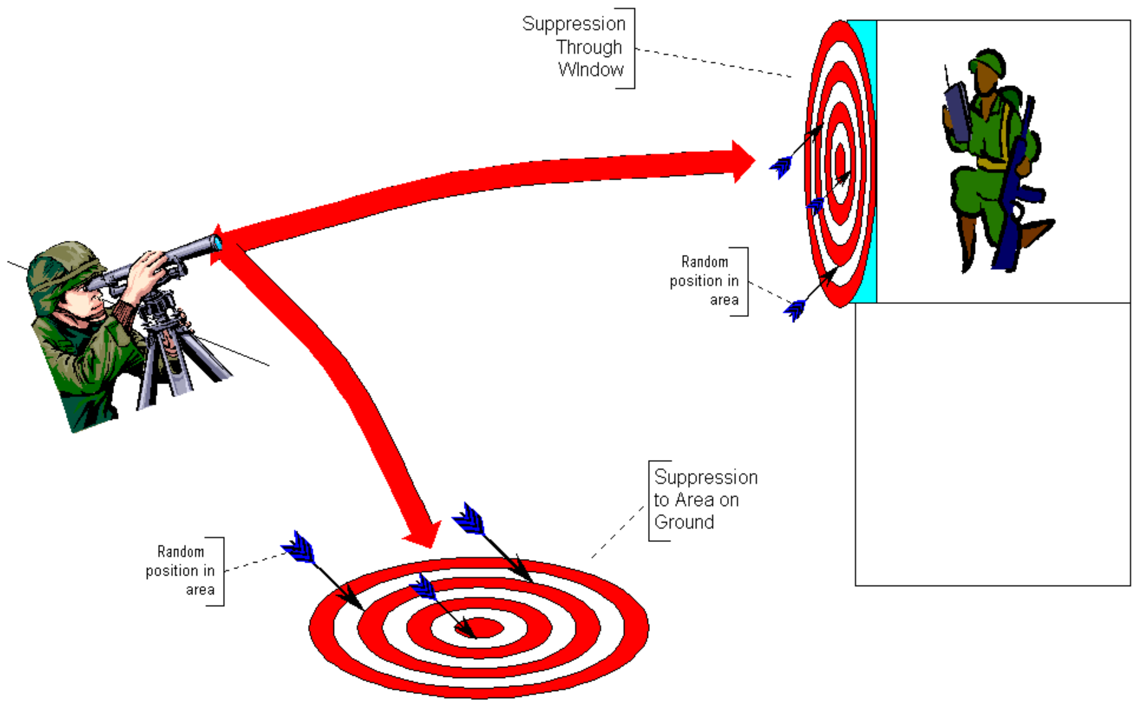

Figure 2 Suppression Fire To An Area

Explosive warheads use a PHPK and create collateral damage to another target. In JCATS 2.0+, the user can play a PHPK high explosive round. It hits the target, then goes to the HE data and computes area effect adjudication. The ED record does not indicate which is the main target and which is the collateral one.

If the target is inside a building, the impact area is drawn vertically (see Figure 2). An impact area close to the window is better. Only those bullets that go through the window can find targets on the inside of the building. The purpose of this is suppression.

Note: floors and ceilings block LOF.

Bullets cannot pass through doors in exterior walls; they can pass through doors in interior walls. Bullets can always pass through windows. Windows are considered to be closed at all times.

Bullets LOF stopped by:

- Dirt
- Exterior walls and doors
- Floors and ceiling in building
- Third interior wall or door
- Fences.

Everything else is a free pass; for example, windows and breaches.

B. Indirect Fire

Artillery is launched usually without seeing the target. It is often called into play with target location by the forward observer (FO). The round travels into parabolic trajectory-lofted shot. However, an FO is not required to fire artillery.

Artillery can fired in three ways:

- Planned indirect, i.e., the person playing the game plans an event using the mouse.
- Triggered by a FO (called direct support with FO)
- Counter battery - Counter battery is new in version 2.3 of JCATS. This option is to shoot based on the trajectory of incoming artillery. The policy is called "shoot and scoot" because the enemy can then pick up the trajectory of the return fire. To play this option, the user must lay down fratricide polygons to keep from shooting his own men. However, the user does not need to play fratricide in JCATS to play counter battery. This mission is a special case of using a FO.

JCATS doesn't care about "flying the shell;" instead, this is what happens (see Figure 3). JCATS does not trace the LOF for the entire path. Rather, it computes where the round should land based on the cross range and the down range of the weapon. Then backing up the minimum of (1000 meters or ¼ of the down range), the model looks for anything that can block the flight during this last part of the flight.

Indirect fire also gives height of burst, which allows a round to explode before hitting the ground. In this case, the model goes to the position of the burst and does LOF from the height of burst (assuming the weapon goes off there) to each possible target on the ground to see if it is hit. This gets the death zone from the exploding weapon.

C. Line of Sight (LOS)

JCATS is a data-driven model written in object oriented code. Objects interact by message passing. Many objects may contribute to a single action as viewed by the user. Figure 4 shows the JCATS generic environmental object and the specific wall object. An environmental object interacts with numerous models, including the physical model, the movement model, the acquisition model, and the interaction model. The portion of each

model invoked by the environmental object depends on the characteristics of the object. For example, the wall object interacts with the wall physical, the linear movement, the linear container acquisition, and the linear container interaction.

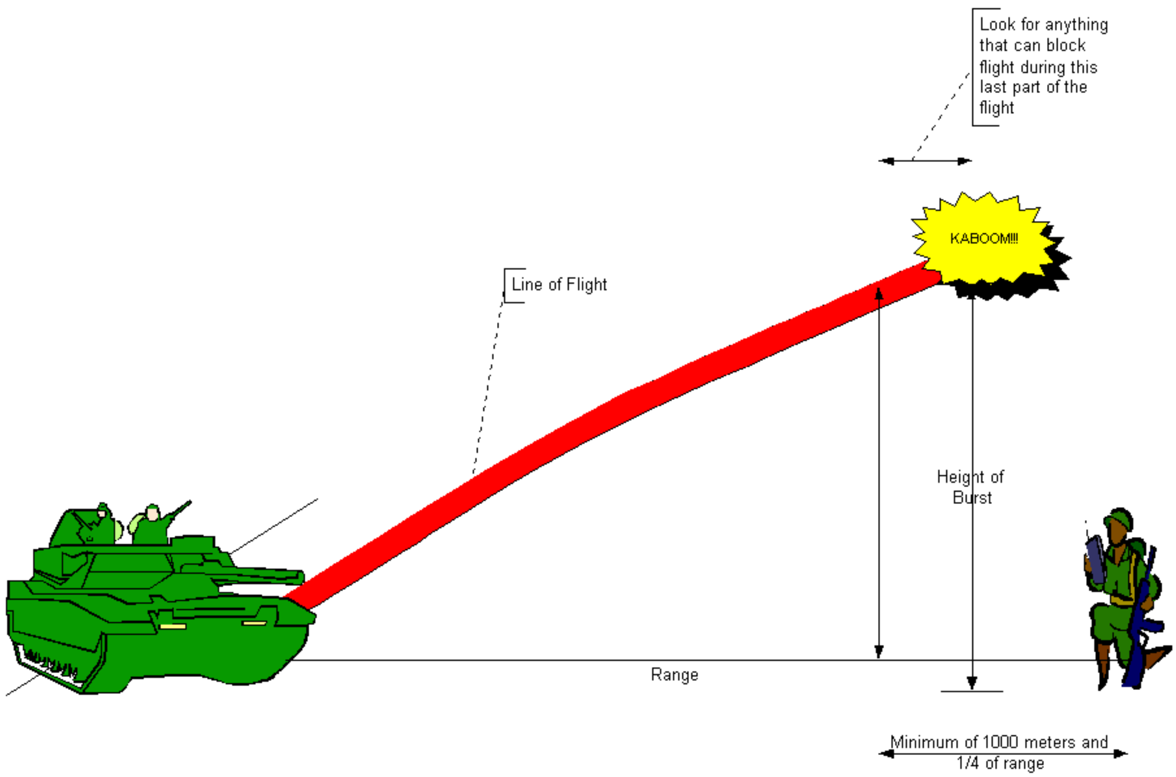

Figure 3 Line of Flight for Indirect Fire

The line of sight calculation begins with an angle created by the line from the seer's head to the foot of the target and the line from the seer's head to the head of the target. Figure 5, 6, and 7 provide three examples of LOS situations.

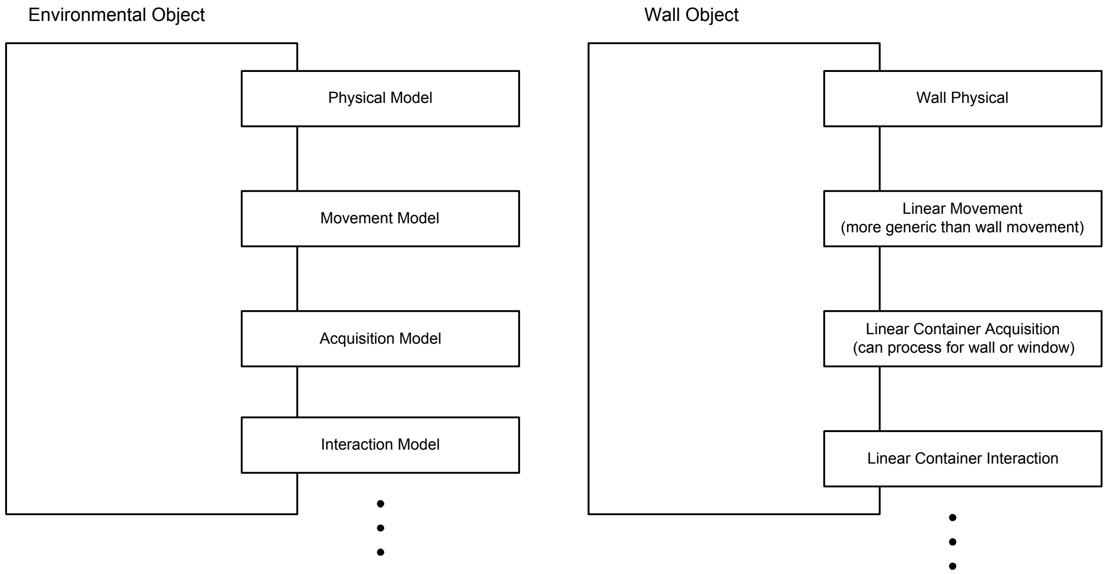

Figure 4 The JCATS Environmental Module

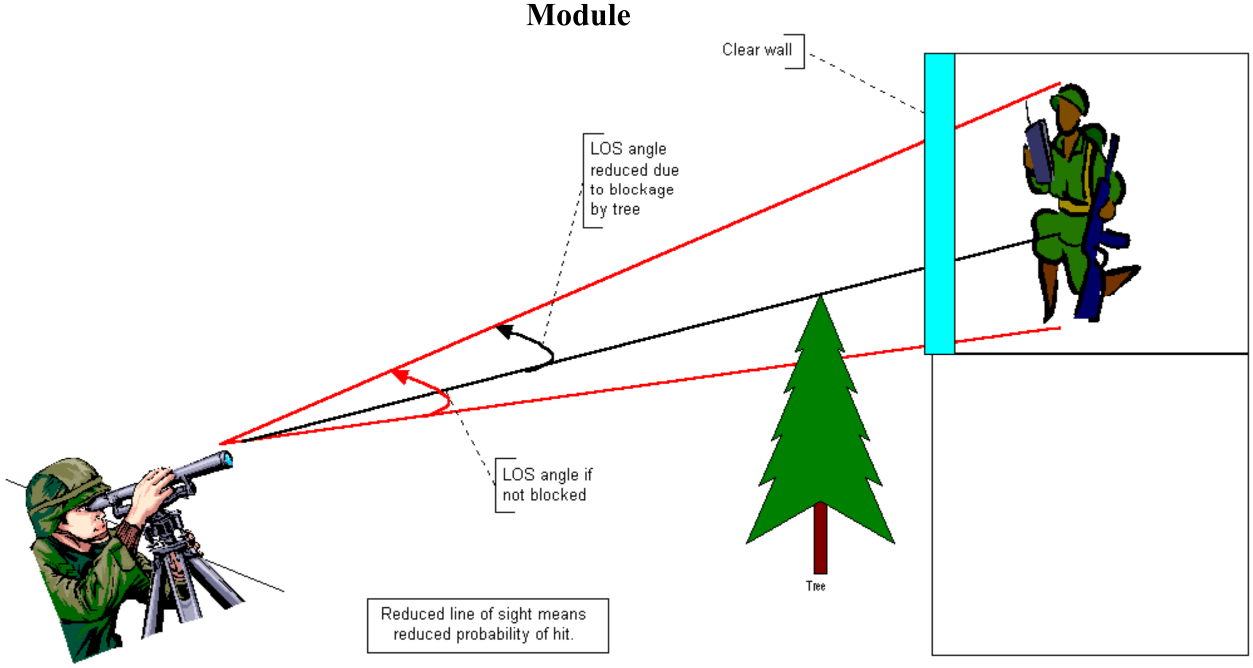

Figure 5 Line of Sight Blocked by Tree

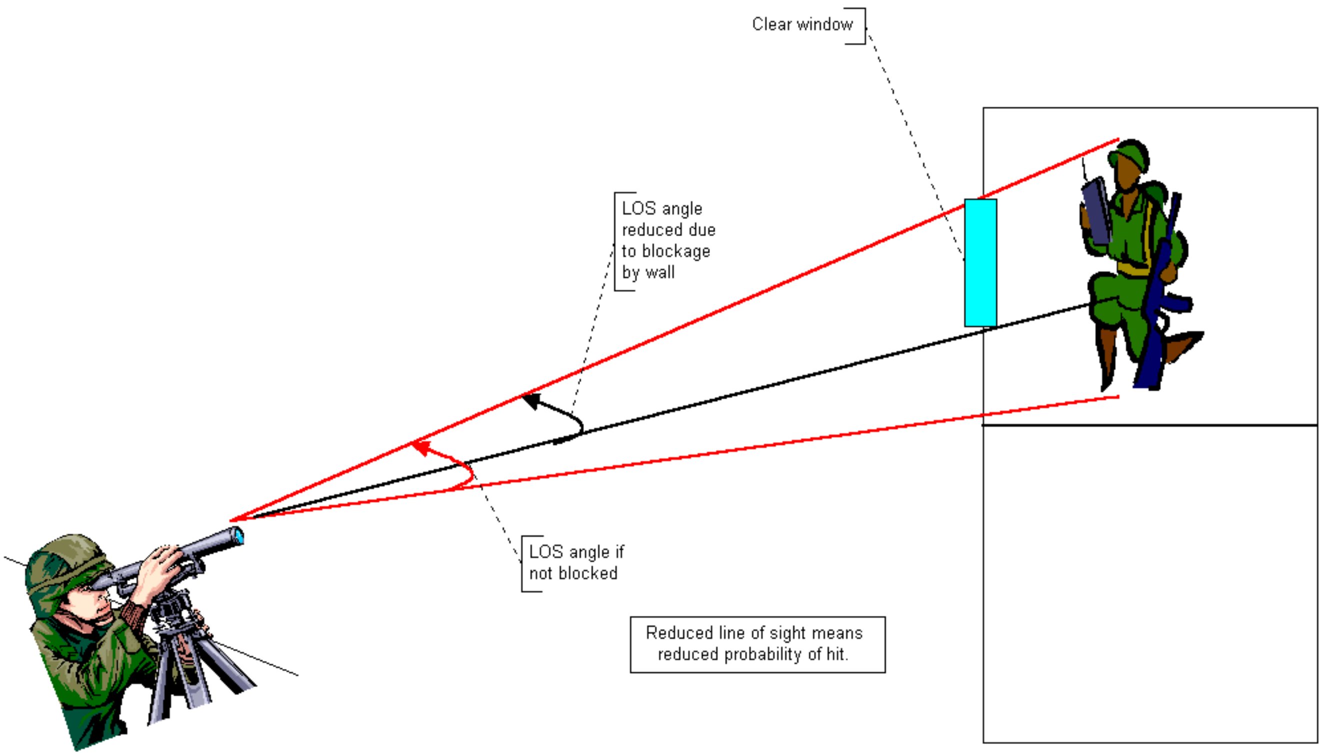

Figure 6 Line of Sight Through Window

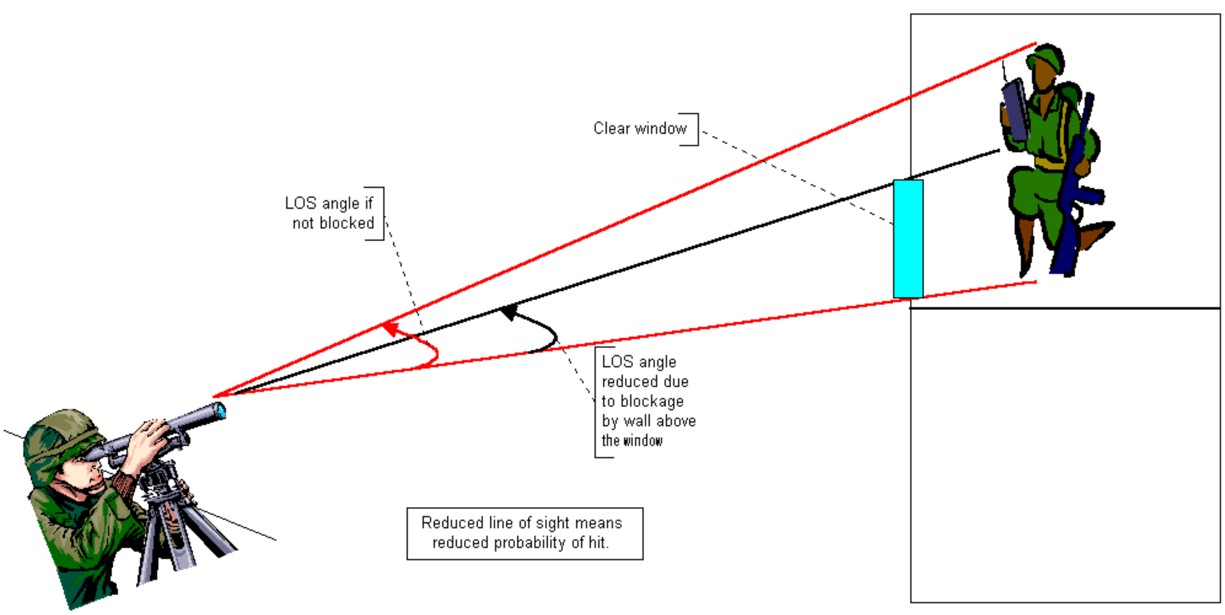

Figure 7 Line of Sight Example With Head Blocked by Exterior Wall

Each environmental object that is between the seer and the target is queried to determine if it affects the angle of the line of sight (see Figure 5). If the object blocks any portion of the line of sight angle, the angle is adjusted to reflect the blockage. Since most environment objects come from the ground up (e.g., berms, walls, buildings), the angle usually is attenuated from the bottom. If a target is located inside a building, the total line of sight angle will be blocked by an external wall, unless the angle goes through a window or a breach in the wall. If the angle goes through a window, the angle may be attenuated from the top by the external wall above the window. If the head of the target cannot be seen, JCATS assumes that the target cannot be acquired and therefore cannot be shot at. This assumption allows a simplification of the LOS calculation because it is not easy to determine if other vital parts of the target are visible. This may not be realistic since soldiers would usually fire at a target if the torso of the target can be seen. While this is a valid concern, most people do not stand taller than window height. However, we suggest that the modeler/scenario builder be aware of this "feature".

Question: What happens to the LOS angle if it goes through two windows, one over the other, but with a portion of exterior wall between them?

If the LOS goes through a window, there is a different check. If it goes through both the window and the wall, the angle is split in two parts and the wall is queried for one portion and the window for the other. If the head ray goes through the window, the window algorithm is used; otherwise, the wall algorithm is used.

In the case that the head ray goes through the window (see Figure 6):

- the man is seen
- the size of the target is reduced by the portion of the target that cannot be seen.
- the probability of hit (PH) is adjusted to reflect the current exposure of the target.

In the case that the head ray does not go through the window (see Figure 7):

- the mid section of the man is seen
- the head is not seen and so the seer cannot shoot at the target.

Note that the location of the head relative to the feet varies according to the posture (standing, crouching, crawling). These sizes are data driven by object type.

One solution to this problem is to create windows that are as high as the heads of most men. This allows the person inside to be seen and for the inside person to better see out.

D. Line of Flight (LOF)

1. Description of LOF Algorithm

The line of flight algorithm was not in the Joint Tactical Simulation (JTS) but was added for JCATS. The JTS always used LOS. The LOF algorithm has great potential for adding capability to the JCATS model beyond its current uses.

The LOF algorithm works as follows:

- Cast one ray from the "shooter" to the "target" (large distance)
- Ask each object along the way if it blocks the ray
- As soon as the ray is blocked, it is assumed that there is an explosion at that point
- Thus the environment lets you know how far you can go.

The LOF algorithm is used for:

- Indirect fire to determine if anything was hit on the way down (see Figure 3)
- Grenade (around the corner)
- Flying shrapnel
- Planned direct fire at an area
- Bullet going beyond the intended target
- Explosion.

The following things can stop a bullet:

1. Dirt (ground)
2. A fence
3. Exterior wall or door
4. The third interior wall or door (a programmer's parameter controls the number of interior walls that a bullet can pass through)
5. Floors and ceilings inside buildings.

NOTE: Some vegetation blocks LOS (if the height of the vegetation is sufficient to fully block the target and the vegetation is opaque), but vegetation does not block LOF. Therefore, a bullet can go through a tree trunk. LOF currently ignores vegetation and only considers elevation (terrain = dirt). There is no difference between grass and tree trunks. In order to modify the code to have vegetation affect the LOF< the following are required: 1) more data to be entered through the terrain editor; 2) determining and implementing an algorithm for relating the LOF degradation data to munition flight and PH degradation. This probably means adding data to each munition also.

NOTE: Bullets are not degraded when they go through an interior wall.

2. LOS Implies LOF

JCATS assumes that if the shooter has LOS, then he has LOF to the target. This occurs for planned direct fire at a target and auto direct fire. LOF is used only if 'fly the bullet'. We do not 'fly the bullet' for planned direct at entity or auto direct fire.

Class of Fire	Subcategory of Fire	Events File Record Type	Use of LOS and LOF
Auto Direct	Direct fire at entity	SD	LOS
	Laser designator	SD/LD	LOS
Auto Indirect	With Forward Observer (FO)	SA/FO	FO has LOS; shooter needs LOF for last of flight
Planned Direct	PDF at area	SD/SF (If hit target (entity) by mistake, create an EA record not an ED record.)	LOF
	PDF at target	SD	LOS
Planned Indirect	Artillery	SA/null	LOF for last of flight

All types of fire can be used against targets inside building, but the munitions must go through the window.

E. Movement

1. Description of Movement Algorithm

The user specifies the time interval for computation of movement for each of the following:

1. dismounted system
2. all air vehicles
3. everything else (all wheeled, track, etc.).

For each system, a stack is maintained of all the things the system is on top of. For example, the system may be on a road, on top of a grassy plain, on top of a flat terrain (see Figure 8). At each time interval, the stack is processed to see how far the system can move. If the system remains on the top in the stack but moves off lower level things, those items are removed from the stack but do not affect the actual distance the system can move. If the system moves off the top during the time interval, movement is processed to the point at which the system leaves that level. If the system can move on that terrain, it continues; if not, the block position is returned. If a system is blocked by the environment, a breach can be set.

During the movement calculation, environmental objects along the movement path must be queried to determine if they block the system. Figure 9 shows how movement is checked as the system encounters or leaves each environmental object: water, hill, tree, or building. If the object is a wall or fence, the system can breach it.

In the movement model, there is a mobility coefficient based on terrain and the moving system.

A person cannot block the movement of any system nor can a person be blocked by another system. It is assumed that a person can always maneuver around other systems.

Vehicles can block other systems provided the flag ‘vehicles block movement’ is turned on. In this case, any system not dismounted can block other systems. Each system has a blocking radius associated with it. These radii are input by the user.

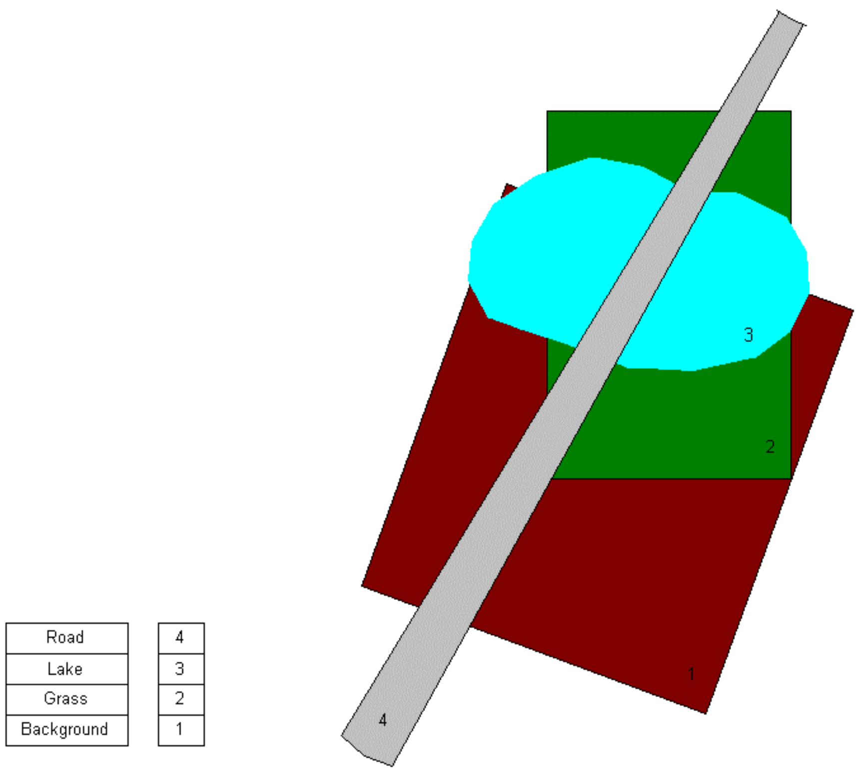

Figure 8 Using Stack To Track Things an Entity Is on Top Of

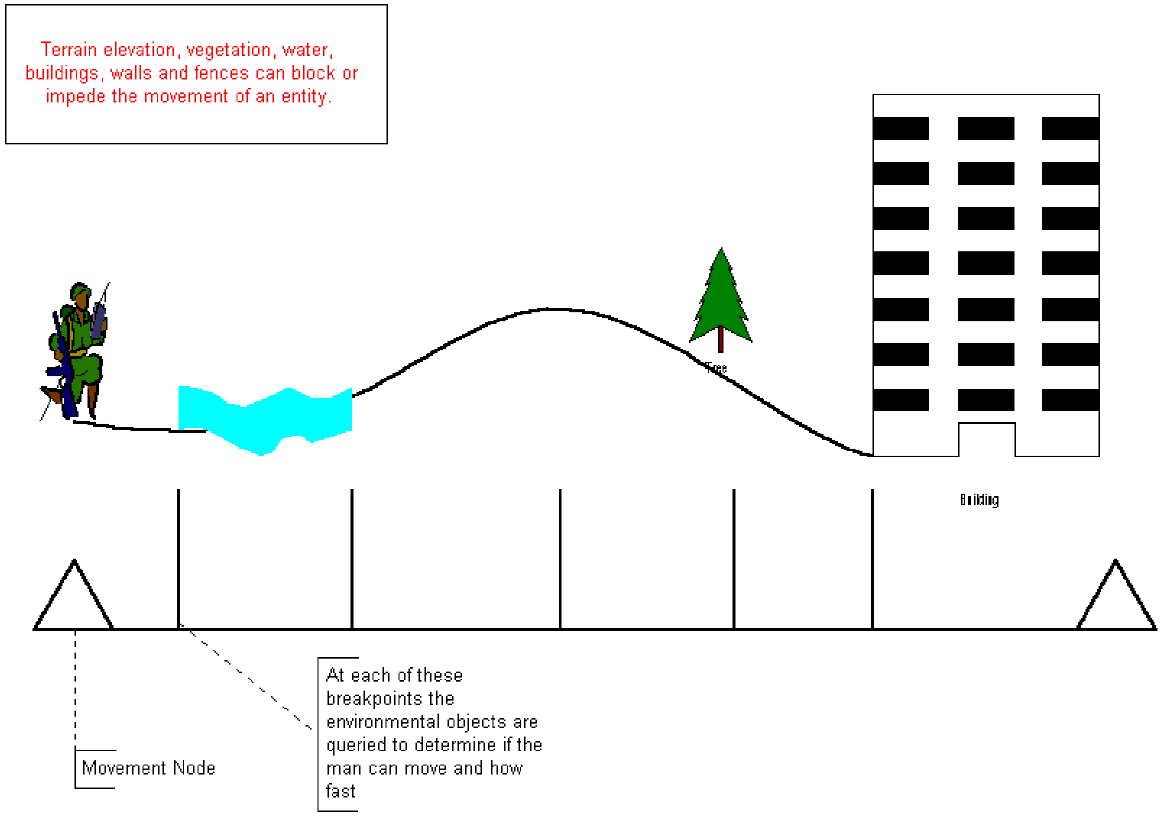

Figure 9. Movement Module

2. Movement and Blocking

The movement model starts moving entities in order by their system ID. An event queue is created for the movement of all systems. For each system, the model checks to see how far the entity can move within the user-specified movement interval. If an entity is blocked, it stops at the blocked position and remains there for the remainder of the time interval. For example, if the time interval is 5 seconds but an entity is blocked after 3 seconds, the entity will stop after 3 seconds and not move for the next 2 seconds. The remaining 2 seconds movement time is lost. At 10 seconds, the model will check to see if the entity can move then. Recall that the flag ‘vehicles can block’ must be turned on to play blocking in JCATS.

In Figure 10, for example, assume that the tanks move in the order of 1, 2, and then 3. Thus in the first time period, tank 1 might move until it is blocked by tank 2. Then tank 2 might move ahead but tank 1 must wait until the next time period to move further. Then tank 3 might move during this first time period but be blocked by tank 2. In the second time period, tank 1 might move to its next movement node and clear the way for tank 3 to move to its next movement node.

If two moving objects start with their blocking radii intersecting, they are allowed to move away from each other (i.e., the angle is 90 degrees or greater). They are not allowed to move toward each other (see top of Figure 11). If blocking is played and an object is stationary and is in the path of another moving object, the second object will never move. In other word, the second object will always be blocked (see lower portion of Figure 11).

Question: How close to a wall can a vehicle get?

Answer: 1 mm. The blocking radius does not affect how close the vehicle can get to the wall. This is may be relevant when dealing with robots, what happens when they approach a wall, curb, etc.

The blocking radius does not impact movement to environment, e.g., buildings. When a vehicle interacts with the environment, it is treated as a point system, i.e., the size of the vehicle does not come into play. Problem: This means that a tank can go between two buildings that are closer together than the width of the tank (see Figure 12).

If a system is stopped by terrain (the slope is too steep) or environment, then a warning message is issued and the user must move the vehicle around the object.

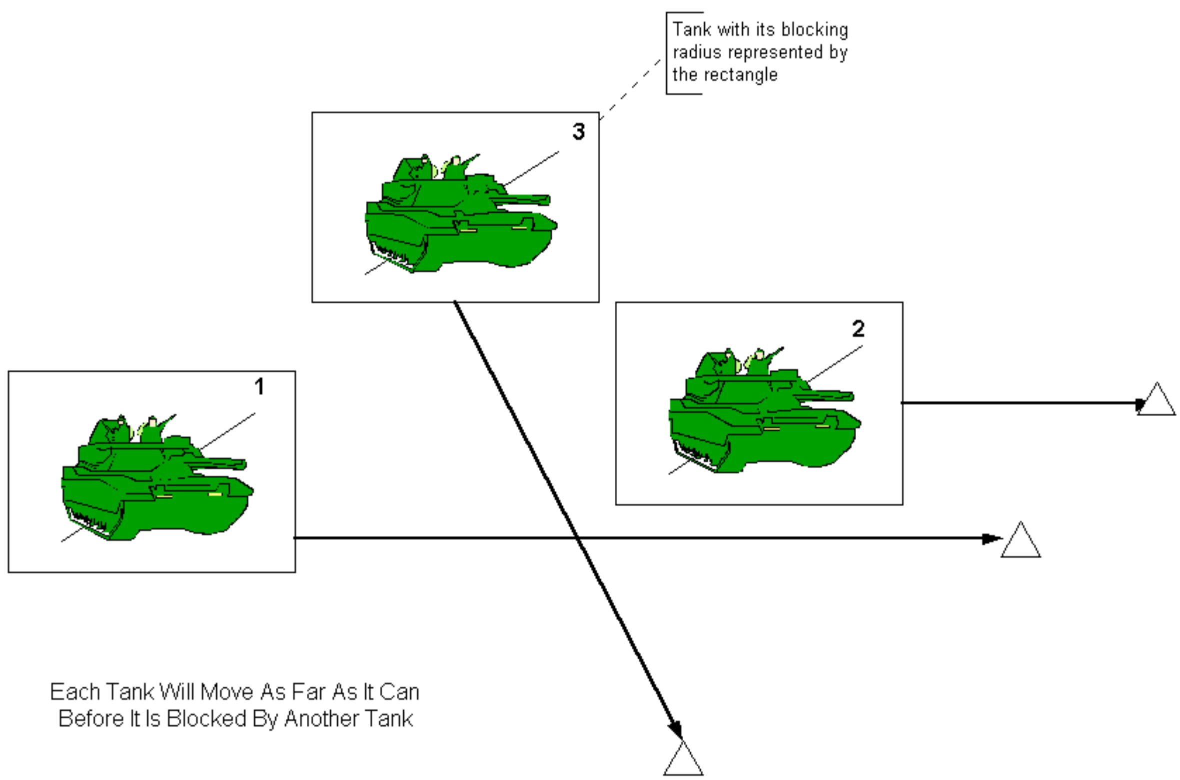

Figure 10 Example of Blocking With Three Moving Tanks

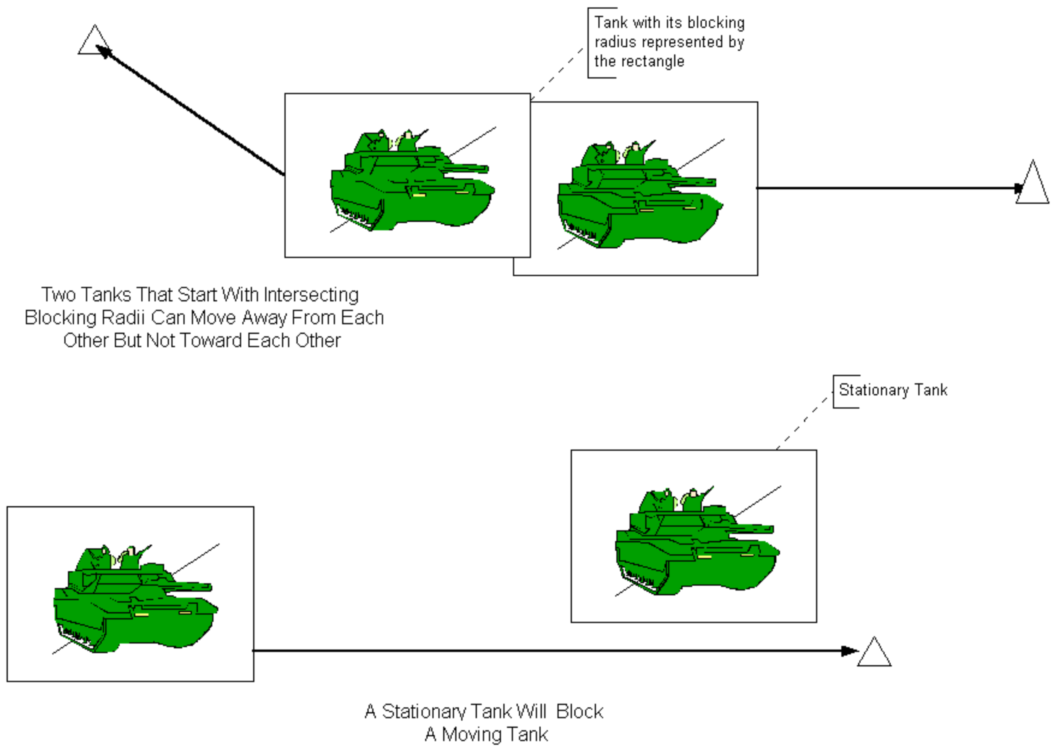

Figure 11 Other Examples of Blocking

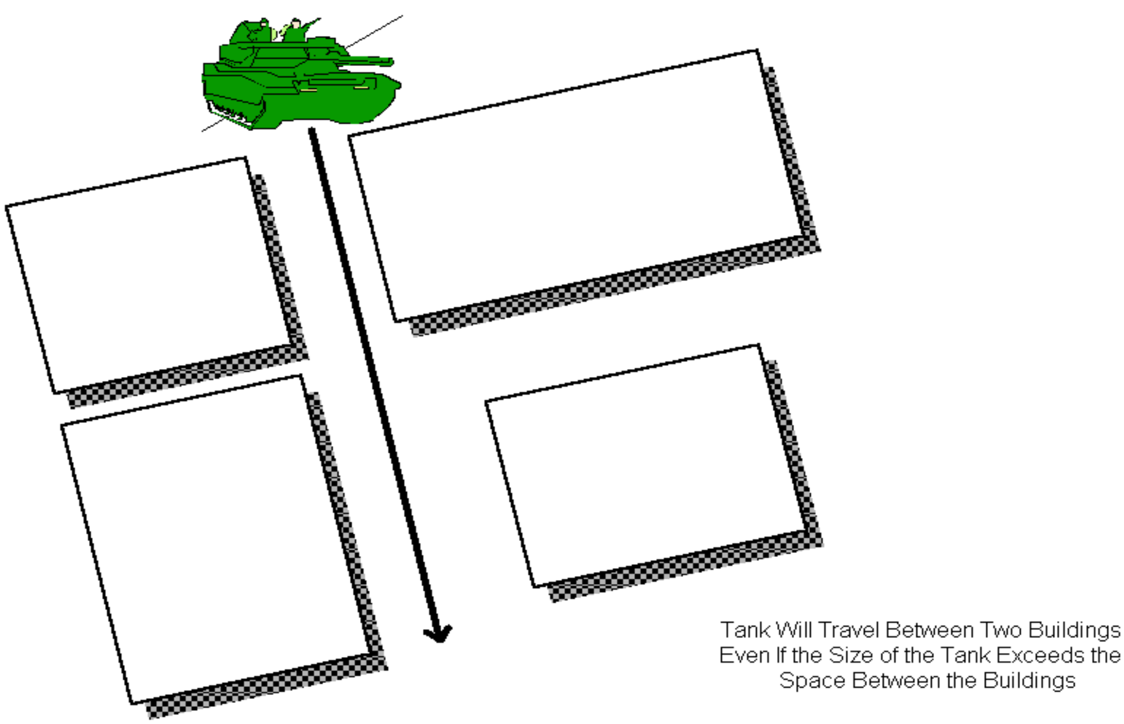

Figure 12 Tank Moving Between Buildings

F. Buildings

1. Entering Buildings

Any system can enter a building on the entry level (see Figure 13). The entry level need not be the first level of the building. A building may have a basement, be built into a hill, or have a ramp up to the entry level. Only people may go to other levels of a building. The floor is not tested for a weight limit.

A ramp is created by creating a road and specifying the elevation of each end point.

2. Rubble

Rubble is only created on the outside of a building. It only affects movement. It does not affect acquisition, LOS, or LOF. The building does not change in any way. The wall is not affected. If the user wishes to simulate a destroyed wall, he can set a breach on that wall. The rubble is put down magically based on the following:

- How close to the wall the system came
- Radius of effect of the munitions
- The type of building.

This determines the radius of the rubble. The rubble can affect all systems. It slows them down, but does not stop them.

G. Protected Areas

Protected areas can be created by fratricide polygons or intel tokens. Fratricide polygons can be defined for No Fire (do not fire in this area) or Free Fire (fire at any target in this area for which level 3 acquisition has been attained). The user specifies the areas and locations of these polygons.

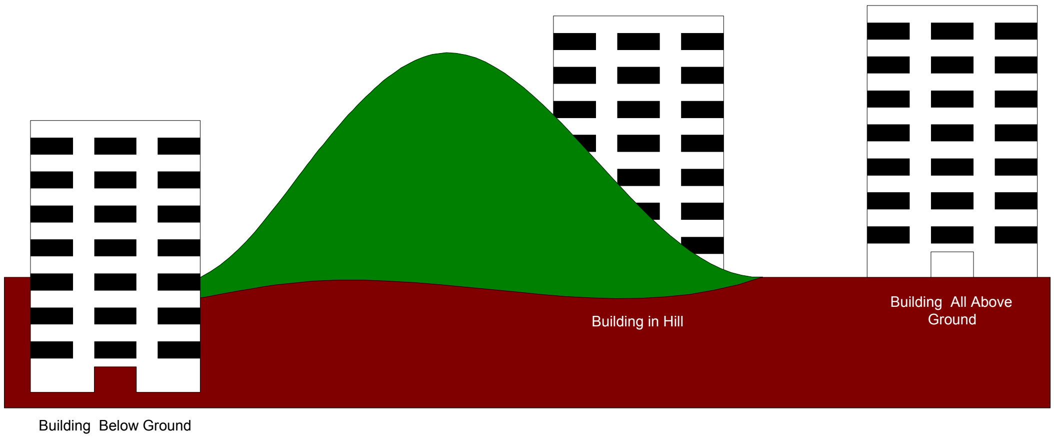

Figure 13 Various Entry Levels for Buildings

Intel tokens are located in enemy areas. If a target is within the area of the token, then there is associated with the token a probability that the target is enemy or friendly. The user determines the location of the intel token but the model determines the size of the token area. The more tokens put down by the user, the more area covered. Intel nodes that are put close together more or less join and cover more area.

H. Penetration

Penetration only allows the passage of an entity, e.g., going through open door. To an LOF, it is not open; thus a bullet cannot pass through an open door. It is assumed that an entity goes through the door quickly and does not keep the door open for the bullet. Thus for LOF targeting, all windows and doors are closed.

I. Breaching

Currently, a system can create a breach in a building. The breach is from the floor to the ceiling of the building level being breached. A system also can breach a fence and this breach goes the full height of the fence. A data flag is set to turn breach on or off. A breach can be for a specific entity that is performing the breach or to create a clear path through a minefield. If the user has not supplied the data for time to breach (the terrain code and moving system by system code), then the system cannot breach or penetrate.

J. Dismounting Radius

Each system that can be mounted must have a dismounting radius. This radius is usually greater than or equal to the sum of the blocking radius for the mounting system and the blocking radius for the carrier. Note that people have no blocking radius. During a simulation when a dismount node is reached by the carrier system, the model will try to dismount all systems. (If the user wishes to dismount only one system he must use the regular dismount mechanism, not a dismount node.) If one system is dismounted using the regular discount mechanism, it goes to the point 180 degrees from the front of the vehicles (or back of boats) and on the dismount radius. If all systems are to dismount, the systems go to ½ the dismount radius. Figure 14 gives an illustration of a tank dismounting troops. **Note**: this description has been corrected here to reflect an error IDA found in JCATS documentation. The user specifies a dismount offset distance in meters, call it d. The systems are spaced out on that circle as follows:

- The first system goes to the point 180 degrees from the front of the tank
- The second system goes to a point d to the left (counter-clockwise) of the first system dismounted
- The third system goes to a point d to the right (clockwise) of the first system dismounted
- Subsequent systems are dismounted on alternating sides of the first system at a distance d from the last system on that side
- Systems may be placed all the way around the circle several times
- If the proposed location for the system to dismount is not valid , the system cannot dismount there. It is then placed at one of the previous slots. Previous slots are tested in reverse order. For example, if a system cannot dismount at the 4th dismount position, the system will try to dismount it at the 3rd and then the 2nd, etc. If a system cannot go to its or any previous dismount points, it will not be dismounted. This is true even if it could have gone to a subsequent dismount point (these are not checked).
- If there are no valid slots into which to dismount, no systems will be dismounted at this node
- Once one of the dismounting entities cannot find a valid dismount point (through all of its choices), all other entities that have not yet been dismounted will stay mounted, but all entities that were dismounted remain so.

The dismounting entity actually moves from a position near the carrier, called the "throw out point," to its dismount point. The throw out point is defined to be on the line of the dismount location at the summation of the carrier and the dismounting systems blocking radius. If that throw out point is further out than the actual place they are to go when dismounting, the dismount point is used. If the carrier is not in a building and the throw out point is in a building, this is considered a bad place to dismount. However, currently, JCATS 2.3 will put the entity on the roof. This will be in version 3.0. If the dismount location is in a building, but the throw out point is not, the dismounting system will try to walk into the building from its throw out point. In terms of the building playing a role in determining dismounting location, only the throw out point matters.

All systems dismount to the back (except boats which dismount from the front), so the user may wish to turn the carrier vehicle before dismounting systems so that the dismounted systems will be able to quickly start on their movement paths.

The dismounting radius is not used for planning but always used for simulation. For mounting, all mounting systems must be within the mounting radius of a carrier in order to mount.

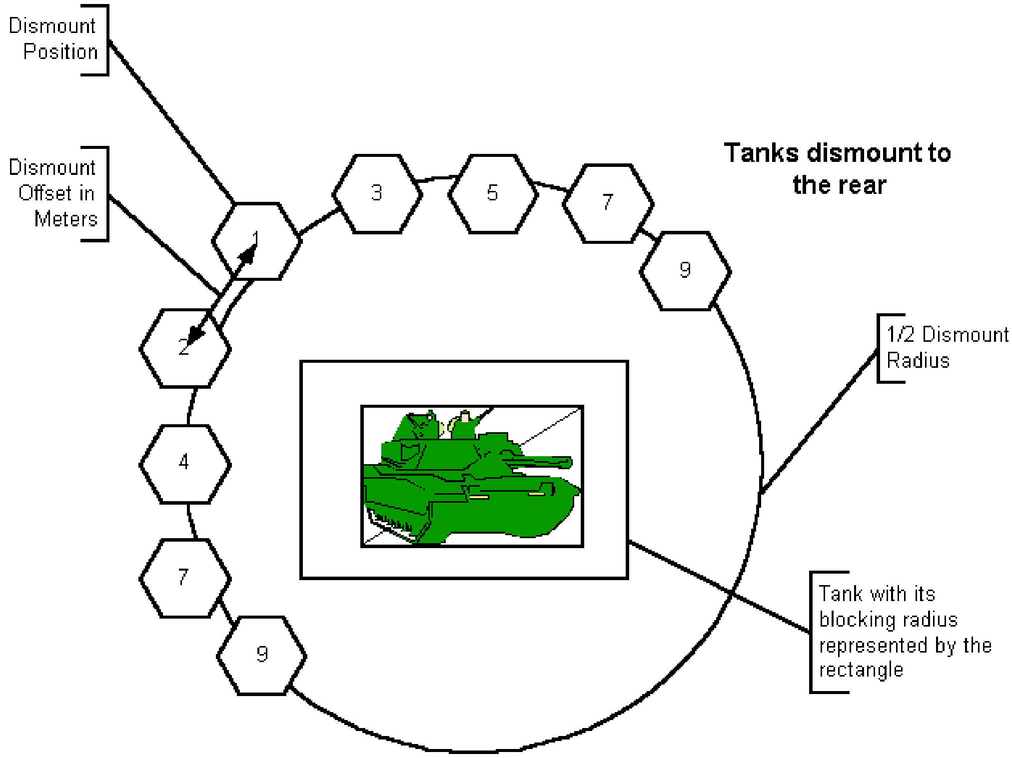

Figure 14 Location of Dismounting Troops When "All" Are Dismounted In One Command

K. Capture and Surrender

In JCATS, you must surrender to be captured.

An entity can be in one of three states:

- Fighting
- Surrendering
- Captured.

When an entity is captured, it is moved to the side and force of the capturing side. The model remembers where the entity was originally so it can go back to that side and force.

From the capture record, one can determine the total number of systems captured by querying for a count of all distinct systems.

L. Casualty and Repair

This feature is mostly an exercise for the user. It may not be good for analysis. A lot of data are required for this feature. A casualty database is required. Repair is useless without casualty play. Repair is performed on people and items.

For each MOB or FP kill of a person, the model does a random draw on the list of types of wounds that can result from the type of kill.

For each of these wounds, data provide the number of minutes a person can survive before getting medical attention. This is for dismounted people only. If the person does not receive this attention, he moves to a KK and a KK record is created.

For all systems the model also needs the following data:

- The amount of time to get repaired, i.e., how long to perform the repair
- How sophisticated a repair station is needed (for example: self repair, medic, hospital).

The user must play the entities. He must move the casualty to a medical facility and then back to the unit after the person recovers.

One could do an analysis, such as how close to put a M.A.S.H. unit.

M. Aggregates

1. Description of Aggregates

Aggregates may not enter buildings. To make members go into buildings they must be de-aggregated.

Because of this rule, if an aggregate moves between buildings and the members are spread out wider than the path between the buildings, the model will jump any members that would appear to be in buildings to the center of the aggregate until the building is past (see Figure 15). This is unrealistic, and on the screen the members skip around. A similar situation occurs when a tank is a member of an aggregate and the movement of the aggregate would put the tank into water that he cannot travel through.

LOS for acquisition is to the center of the aggregate. If members are not in an aggregate, then the seer gets a chance to acquire each member of the aggregate.

For targeting, LOS is to the location of the system within the aggregate.

If all members are moved to center of the aggregate, then all could be killed with one weapon that hits at this point.

The model loses fidelity when aggregation is used, but the model runs faster. It is a tradeoff between performance and "puckability." Aggregation is good for large campaigns, flat terrains, and for bringing troops to the edge of an urban combat area.

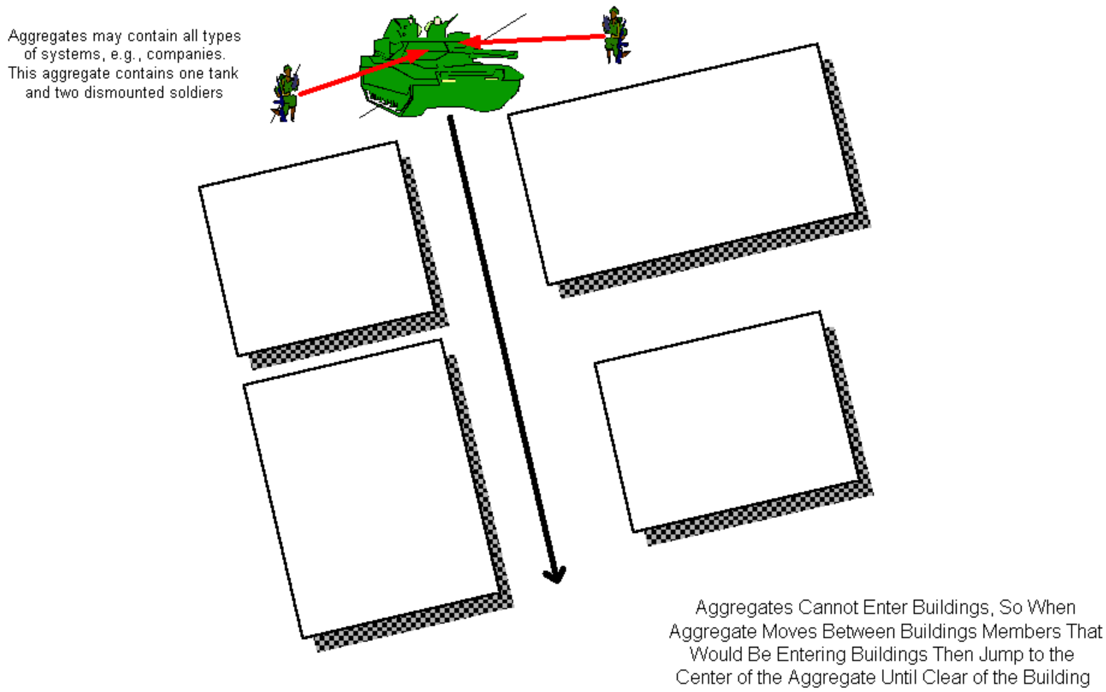

Figure 15. Movement of Aggregates

2. Aggregates and Acquisition

The sum of the optical dimensions of the aggregate make detection easy, i.e., the aggregate is easy to see. Getting to the classification, recognition, or identification levels is much harder. The largest element of the aggregate is used for identification.

LLNL ran tests on aggregates. The following conclusions were reached:

- When two large aggregates were run, the run was much faster but the output was large. Also, the target list was very large.
- Using three levels of aggregation is optimal for a balance between run time and size of output (e.g., squad to platoon to company).
- When one sees an aggregate, he sees all the elements. This creates a large target list.

There is a difference between a station view of aggregation vs. a command structure to create aggregation. For MOUT, it is reasonable to use aggregation to transport troops to the urban area and then de-aggregate them.

Possible modifications to JCATS:

- Allow the user to play limit on sensors coalescence (i.e., merging together), then individual units would be on LOS.

- Do not let total aggregate be the target. However, it is then hard to decide what the target is and what should be reported.

Currently, formed aggregates subsume the acquisition of their components. Sensors are coalesced at the center of the aggregate. This provides significant run-time economy; for example, an aggregate of 100 troops coalesces all 100 eye-ball-pair sensors into a single eye-ball-pair sensor operated by the aggregate, achieving a 100X reduction in LOS/sensor calculations. Unfortunately, this is an approximation that quickly breaks down as the physical span of the aggregate (i.e., the span of its parts) gets anywhere near the range of the sensors. In the worst case, an aggregate cannot see even to its own physical boundary!

What we were talking about is limiting the coalescence of sensors to only those cases where the aggregate span is small compared to the smallest sensor range. From a coding point of view, this is easy in the simulator, but will take some work on the part of the client.

Also, currently formed aggregates are acquired as nothing or the full aggregate. Probably (especially for aggregates with large physical spans), the components should be individually acquired. This will take some work on the simulator, and significant work on the client if there is any requirement to “declutter” the screen by replacing the components of a fully acquired enemy aggregate with just the aggregate itself. And it will take a bit more work if there is a requirement to decompose an enemy aggregate into its components.

N. Interrupting Simulations

When interrupting a simulation, i.e., stopping and restarting a simulation, care must be taken to make ensure that all needed data are saved.

What is saved:

- Movement paths
- State of completed engagements.

Not saved:

- Current engagements
- Artillery missions
- Acquisitions
- If mission is incomplete (e.g., 2 out of 5 done), all are lost
- Weapon recovery (reverts back to the dead body).

The best policy is to run to a break point in the battle, save the plan, shut down the simulation, and restart. For short scenarios it is best to not interrupt.

O. Event File

The user can transfer entities between forces or task forces, but not side. Also, units will change sides upon capture and again upon reentry to their original owners.

The station number is a constant. Each system has a number of stations from which a weapon can be fired. Under weapon recovery, a system may get an extra station. The Recover Weapon function recovers a weapon station including weapon(s), ammo, and sensor.

In the event file, JCATS creates a default unit name as the class name underscore the system ID. Note that the system ID is a permanent ID assigned to each system. It is unique over all systems, not just within a side. Thus, within a study the IDs are constant over all scenarios and runs.

All members in an aggregate must be in the same task force.

In version 1.2 of JCATS, MK records (mounted kills) were not generated. In version 2.3 they are generated but they are not reflected in the AWS file kills.dat. A mount kill is always a critical kill (KK) and only occurs when the carrier incurs a KK. To determine the shooter for the mounted system, it is necessary to use the clock time and carrier ID on the MK record to find the AK or DK record for the carrier and then use that shooter data.

In JCATS, crew are usually personnel required to operate the carrier. Most users of JCATS do not use the crew to restrict the use of equipment. However, police and security use crew in this manner.

The Acquisition record(AQ) now has range included.

On the IA record, the ‘effect ‘ field gives the munitions type: ball/point/ICM/HE.

On ID record, other types of kills are from old files, no longer used.

MISC files give the system type ID.

In a JCATS scenario, keep the unit type name to a maximum of 12 characters so that it can be used to group systems. The model creates default unit names as unittypename_1, etc.

The model does not report who lays down a minefield. Minefields are created when a barrier is laid down. However, this record does not contain the side that creates the barrier.

Blue can kill Blue:

- Fratricide (ED and ID records when shooter and target are same side)
- HE
- Minefields.

AWS treats the acquisition records (AQ) as start of acquisition when effect equals ‘AQ’ and as end of acquisition when effect equals ‘BR’, ‘BUC’ or ‘BUN’.

AWS does not do mount kills but does do minefields. We do not know what side is given to the shooter for minefields.

In AWS it is an aviation kill if the shooter is a fixed wing aircraft as indicated in the datspot file. Otherwise, it is a regular kill.

In AWS the IA round type impact = 0 for normal and 200 for PGM. This value is stored in the INT2 field in the events.dat file.

On the event file, SELEM and TELEM are always 0.

There is a new capability called Partial Damage.

A system can be hit with numerous kills of all types. There is no cumulative effect from kills. There is no relationship between kinds of kills. The model does not report a change in state on the last record. If there is a MOB kill and later a FP kill, two records are created and no MFP kill is reported. To get the final state of each system after all kills, a post-processor would have to combine the data appropriately.

There are three runtime game parameters in the Vista editor that allow the modification of the priority for attacking a target again when the previous attack was MOB, FP, or MFP, respectively. The parameter is a target_partial_kill_weight having a value between 0 and 100 percent. The priority of a target is multiplied by this value to obtain a new priority after the target has been MOB, FP, or MAF killed. If the parameter is set to 100 percent, the target retains its original priority. If the parameter is set to 0 percent, the target has no priority and will not be attacked again. If the parameter is set to 75 percent, the new priority is 75 percent of the normal priority. Priority values for targets are relative, with higher priority-valued targets being attacked first. For example, the model will attempt to attack a target with a priority value of 90 before one with a value of 80.

P. The Acquire Model

The versions of JCATS examined for this V&V do not use the ACQUIRE model, but the algorithm used by JCATS does, like ACQUIRE, account for two dimensions. Note: The newest version of JCATS, version 4.0 (released October 2002) will include the ACQUIRE model.

For vehicles:

The ACQUIRE model calculates the presented area of the target to the sensor based on the angle of observation. It uses the square root of the presented area as the linear dimension for computing the milliradians of subtended arc across the target. The sensor function provides the bars-of-resolution per milliradian at the calculated target contrast, and so the product of these two gives the bars resolved on the target. JCATS ignores the angle of observation and uses the average presented area of the target over the full 360 degree possible viewing angles. In this way it differs from ACQUIRE. JCATS then uses the square root of the presented area as the linear dimension for computing the milliradians of subtended arc across the target. The sensor function provides the bars-of-resolution per milliradian at the calculated target contrast, and so the product of these two gives the bars resolved on the target. In these ways, JCATS is identical to ACQUIRE.

For Dismounted:

We do not know how ACQUIRE handles people. JCATS computes the linear dimension to be the silhouette area (for the current posture) divided by the height (for the current posture).

JCATS classification levels are different from the Night Vision & Electro-Optics Laboratory (NVEOL) classification levels.

In JCATS the levels of acquisition are:

Detection – something is seen

- Classification – e.g., tracked vehicle
- Recognition – e.g., Bradley
- Identification – e.g., US Bradley

The Acquire model refines the classification level (gross and fine), but does not have an identification level. In the Acquire model, the levels of acquisition are:

1. Detection
2. Gross classification
3. Fine classification
4. Recognition.

Q. Bullet Proof Glass

There are two workarounds for modeling bulletproof glass:

1. Make an exterior door or wall that one can see through. Then the LOS is not blocked but the LOF is blocked.
2. Use three transparent interior walls very close together to stop LOF through interior of building. The problem here is that the shooter keeps trying to shoot through the glass.

Note: These two workarounds do not work for all types of fire missions. See test results under the Miscellaneous category in the main V&V report.

R. Defilade in Buildings

JCATS does not play defilade inside buildings. A workaround may be to create engineering obstacles inside the building to create defilade effects. There are problems with this approach.

Defilade is used as a lookup index in the PH tables to make the target smaller for acquisition. An entity cannot be completely hidden. There is always a portion of the target exposed, even in full defilade. In other words, associated with each level of defilade is a height of exposure. The minimum height that can be exposed is that height associated with full defilade.

Engineering objects use the defilade heights. However, engineering objects do not affect LOS. One might use a half wall with window on top as a workaround for defilade in buildings. A person must be standing in an object for it to affect LOS. For example, if there is a stack of sandbags between two people, neither of which are in the stack, the two people can 'see' each other no matter how high the stack.

S. Automatic Route Planning

JCATS has no automatic route planning.

T. Fatigue and Stress

JCATS does not model core body temperature (heat) or differences in the way the body functions under stress (arms impaired, etc.). However, it does model the effects on actions.

JCATS requires energy to do a task. Rest restores an amount of energy. The effect of lack of energy is to increase the time to perform a task and to change the accuracy with which the task is performed. The following tasks are modified:

- Speed
- PH
- Acquisition
- PH degradation
- Weapon repair time
- Weapon laydown.

U. Dynamic Terrain

JCATS allows the addition of engineering objects such as laying down mines, craters, ditches, and foxholes. The sea height can also be changed for boats. JCATS does not allow the user to move dirt in real time. The user can change the terrain offline using the terrain builder, but then he must restart the model with the new terrain data.

V. MOUT Problems

There are two areas of concern for MOUT:

- Rubble
- Road width not being played.

ANNEX B

MOUT CAPABILITIES

The chart on the following pages lists a set of MOUT modeling capabilities/ requirements; it also matches up JCATS capabilities with these requirements and describes any special features or limitations with JCATS with respect to a specific requirement. The MOUT modeling requirements as well as the JCATS assessment were performed by IDA for a separate task undertaken for the Joint Staff. That work served not only as a stimulus to conducting a MOUT V&V effort of JCATS, but also the resultant set of MOUT modeling requirements was used to select and prioritize the JCATS algorithms to be examined during the Verification phase of the MOUT V&V.

Model Capabilities	In JCATS	Caveats for JCATS
1. Dismounted Combatant		
□ Postures:		
□ standing	x	
□ kneeling	x	Kneeling/crouching.
□ crawling		Can move at slow, medium or fast pace.
□ prone	x	
□ foxhole	x	
□ running		Can move at slow, medium or fast pace.
□ walking		Can move at slow, medium or fast pace.
□ Wounding:		
□ KIA (Killed in Action)	x	Catastrophic or firepower kills of soldiers modeled.
□ mobility	x	
□ casualty evacuation	x	In V2.0 release and later.
□ Tie-in to Terrain:		
□ 1-meter digitized	x	DTED level 5 in V2.0 release and later.
□ covered concealment		
□ interactive with rubble fences, buildings, etc.		
□ Suppression: human response		Conventional and secondary suppression can be modeled.
□ Interaction with Vehicles:		
□ mount/dismount	x	Aboard vehicles, ships/boats, aircraft.
□ ride atop/outside		
□ Weapons Effects:		
□ infantry small arms	x	
□ automatic weapons	x	
□ area and directed fire of direct-fire weapons	x	
□ hand grenades	x	
□ mines	x	Anti-personnel & anti-personnel mines.
□ claymores	x	Anti-personnel.
□ Movement:		
□ impediments to dismounted movement	x	Models water, buildings, fences.
□ automated taking cover	x	If in defilade.
□ automated “pop-up” from cover	x	

□ automated attack movements	x	For the most part, not automatic; however, behavior module in V3.0 release and later allows some automated behaviors.
□ Sensing and Detection:		
□ sound ranging and direction	x	
□ naked eye	x	
□ effects of vision devices	x	Thermal and optical sensors are modeled.
□ smoke and obscurants	x	
□ Fatigue and Stress:	x	Heat and stress affect both speed and actions.
□ heat		
□ body function		
□ effects on actions		
2. Combat Inside/Around Buildings		
□ Shell of Building:		
□ blocks movement	x	
□ blocks LOS (Line of Sight)	x	
□ Rooms and Floors:		
□ shape/structure	x	Can model multiple floors, interior walls, doors, windows, rooftops.
□ furniture	x	Furniture not explicitly modeld, but in V3.0 release can create features in rooms that block LOS (similar to furniture).
□ Stair Wells and Elevators	--	No stairwells/elevators. However, in V3.0 release ramp-like structures can be created.
□ Exterior/Interior Wall Materials and Effects on Munitions/Demolitions		In V1.1 release and later, an entity inside a building is provided some protection by the walls against an explosion outside the building. However, the entity is not protected from explosion outside if it is also outside or if both the entity and the explosion are inside.
□ Rubbling:		
□ effects mobility	x	
□ effects on building LOS	x	Standing buildings do affect LOS.
□ effects of rubbling on LOS		Rubble does not affect LOS.
□ Windows and Doors:		
□ pass through	x	
□ see through	x	
□ shoot through	x	
□ reductions to LOS and firing projectiles	x	

□ Interior Illumination/Temperature		
□ External Walls and Windows: firing in and out	x	
□ Texture to External Buildings	x	Also models exterior illumination, including artificial lighting from spotlights.
□ Covered Concealment Around Buildings	x	
□ Use of UGV (Unmanned Ground Vehicle) Inside Buildings	x	
3. Combined Arms/Joint Force		
□ Indirect Fire:		Also models forward observers.
□ modes of employment		There does not appear to be a limit on the angle through which a gun can move.
high angle	x	
direct fire	x	
□ laser guided	x	
□ laser spotting	x	
□ missiles	x	
□ smart munitions	x	
□ smart mortars	x	
□ Aircraft:		
□ close air support		
cannon	x	
bombs	x	
rockets	x	
missiles	x	
smart munitions	x	
forward observer	x	
laser guided	x	
laser spotting	x	
□ helicopters		
cannon	x	
bombs	x	
rockets	x	
missiles	x	
smart munitions	x	

forward observers	x	
laser guided	x	
laser spotting	x	
□ UAVs (Unmanned Aerial Vehicle)		
flight	x	
sensors	x	Variety of sensors modeled.
data stream		
munitions	x	Variety of munitions modeled.
□ interrupt LOS/line of fire trajectory	x	Buildings block an aircraft's LOS and flight of trajectory of weapons.
□ aircraft vulnerability		
□ air defense (radar, missiles, artillery)		
□ Armor:		Models variety of armored vehicles.
□ capability		
□ vulnerability		
□ Naval Forces:		V1.1 release and later models submarines; V2.0 release and later models aircraft carriers.
□ munitions		
shells		
missiles		
□ targeting		
forward observer		
laser guided		
laser spotting		
4. C4ISR		
□ Command and Control:		
□ automated and/or interactive	x	Interactive, provided by operator.
□ planning assistance to operator	x	Some assistance provided, e.g. CAC files and emails between workstations.
□ Communications:	--	Not modeled between entities, except "perfect" communication between forward observer and indirect units.
□ voice		
□ hand signals		
□ radio		
□ transmission reliability		
□ networking		

□ digital or analogue		
□ data and/or voice		
□ Integration of Air/Space Intelligence Means		
□ Use of Sensors for Surveillance:		
□ air	x	
□ ground	x	
□ national intelligence		
□ IR (Infra-red)	x	
□ thermal	x	
□ radar	x	
□ sound	x	
□ visual	x	
□ Integration of Joint C4ISR		
5. Rules of Engagement (ROE)		
□ Number of Sides:	x	Up to 36 sides.
□ can they be made:		
friendly	x	
enemy	x	
neutral	x	
neutral/hostile	x	
neutral/friendly	x	
□ can this be done interactively	x	Can be changed during the game by a controller.
□ Hold Fire Capability:		
□ no fire	x	For individual entities.
□ return-fire only	x	For individual entities.
□ pre-emptive	x	For individual entities.
□ No-Pursuit/No-Fire Zones	--	Two types of areas can be created using intelligence tokens: No Fire Areas (NFA) and Free Fire Areas (FFA). In NFAs, systems and aggregations are more likely to consider unidentifiable entities friends. The reverse is true for FFAs: systems and aggregations are more likely to consider unidentifiable entities enemies. No-pursuit and no-fire zones, however, cannot be modeled.
□ Limitations on Munitions/Platform Use	x	Through target mission priority tables or through PH/PKs set close to 0 or weapons designated to be used at specific ranges.
□ Conditions of Non-Lethal Use	x	

□ Prisoners of War (POW)	x	Also surrender of entities.
□ Resupply of Neutrals	x	
6. Subterranean		In V2.0 release and later.
□ Construction of Underground Structures	x	Basements, tunnels, subways can be modeled. No special effects of munitions inside underground structures are modeled.
□ Movement:		
□ speed	x	
□ canalization	x	
□ Sensing:		
□ visual	x	
□ light	x	
□ sound	x	
□ other sensors		
□ Communications Effects:		
□ voice		
□ radio		
□ trailing land line		
□ Weapons/Munitions Effects:		
□ small arms rounds		
□ demolition		
□ grenades		
□ mines		
□ Environmental Effects:		
□ absorb/reflect shock from explosions		
□ reflect projectiles		
□ echo		
□ concussion		
□ collapse		
□ lighting		
□ temperature/humidity		
7. Non-Lethal and OOTW		

□ Incapacitate Individuals	x	Individuals can be incapacitated by munitions like rubber bullets, bean-bags, flash-bang munitions, and pepper-spray. These munitions may be given a small probability of kill, though the focus is on their suppressive effects. No means for disabling vehicles currently exists in the model.
□ Disable Vehicles		
□ Simulate Weapons Effects:		
□ gas	x	
□ nets		
□ foam		
□ "soft" projectiles		
sponge grenades	x	
rubber bullets	x	
wooden blocks	x	
□ tazers		
□ mace/pepper spray	x	
8. Treatment of Vehicles and Aircraft		
□ Hulks:		
□ creation	x	
□ stay on screen	x	
□ effect on LOS	x	
□ effect on movement	x	
□ Ground Movement Sensitive to:		
□ space between buildings	x	
□ ground support (trafficability)	x	
□ road width	x	
□ Air Movement:		
□ curve smoothing within 6 degrees of freedom	x	
□ landing and take-off	x	
□ helicopter movement (hover, vertical, etc.)	x	
□ Modeling of Platform:		
□ placement of sensors and weapons		
□ articulated parts		

□ limitation on elevation/depression angle for weapons/sensors		
□ number of sensors/weapons allowed		
□ crew modeled	x	
9. Terrain		
□ Dynamic Terrain	x	
□ Terrain Resolution	x	
□ Buildings	x	
□ Subterranean Features	x	
□ Cultural Features	x	
□ Vegetation	x	Also bodies of water (oceans, lakes, rivers) and roads are modeled.
10. Technical Considerations		
□ Programming Language	x	Written in C and C++.
□ Object Oriented	x	
□ Computers Employed:		
□ type		Runs on a suite of Hewlett Packard 9000 Series 700 UNIX workstations or on a suite of Sun Solaris workstations. The version 2.0 release will include a version of the simulation that runs on a PC running the Linux operating system.
□ number		One or more.
□ DIS/HLA Compliant	x	The DIS bridge is limited in that it cannot handle more than three sides or aggregations of entities, but HLA compliance does not impose these restrictions.
□ Interactive vs. Closed		Interactive
□ Immersiveness		
□ Realtime or Faster or Slower	x	
□ Data Retrieval/Data Tools	x	
□ Input/Control of Variables/Over-Ride	x	
□ Built for Training or Analysis	x	
□ Ease of Expansion/Modification	x	Configuration control makes for accuracy but may slow process of making changes.

APPENDIX C

JCATS FIRE MISSION DESCRIPTIONS

Institute for Defense Analyses

Figure 1 Setting Up Various Types of JCATS Fire Missions

Type Attack	Type Target	Shooter Attributes			Munition Capability	Coordinator Attributes		Must Acquire Target	Method of Assessment
		Shoot[1]	Hold Fire	Direct Support (DS)[2]		Forward Observer (FO)	Laser Designator (LD)		
Auto Direct Fire	Target	ON	OFF		Auto DF			Shooter must acquire	PHPK
Planned DF at Target[3]	Target		OFF		Planned DF			Shooter must acquire	PHPK
Planned DF at Area (Suppres-sive Fire)	Area		OFF		Planned DF				Fly the bullet
Planned Indirect Fire	Area		OFF		Planned IF				Area effect
Direct Support (DS) with Forward Observer (FO)[4]	Area	OFF	OFF	ON	DS.	ON		FO must acquire	Area effect
Direct Support (DS) with Laser Designator (LD)[5]	Target	OFF	OFF	ON	DS.		ON	LD must acquire	PHPK

[1] Shoot must be ON for auto direct fire but is not considered for other missions. If Shoot is ON for an FO, the FO will perform auto direct fire while the shooter also provides DS. If Shoot is ON and Hold Fire is ON, the shooter will not fire unless it acquires the target and the target fires first.
[2] Mission is in support of another system which must call for the attack using a Forward Observer or Laser Designator. The FO and LD coordinators must be in the same Force as the shooter that is called on to provide DS.
[3] Planned DF at Target missions must be created during the JCATS simulation, and after the shooter has acquired the target. A system cannot fire at the target in this type of mission unless that system itself has acquired the target.
[4] FO looks inside the same task force for a DS shooter, but if one is not found it looks in another task force for a qualified shooter.
[5] LD must be in same task force as DS shooter.

METHODS OF ASSESSMENT

1. **PHPK Munition**
    - Acquire the target
    - Get PD for target
    - Adjust PD for partial blockage using PH tables and adjustment equation

2. **"Fly the Bullet"**
    - Follow the path of the bullet
    - If the path intercepts a target it hits the target.
    - If the path does not intercept the target then it does not hit the target.
    - There is no partial blockage in this case.
    - If a munition is "trackable" in the JCATS database and the parameter "track missed shots" is turned on, then JCATS will fly the bullet for a Direct Fire mission when the munition misses the intended target. This will then determine what other targets the munition may hit after the miss. This feature greatly increases the computations required in the model.

3. **Area Effect**
    - Weapon is fired at an aim point.
    - An impact point is determined. The LOF of the weapon can be blocked by terrain, buildings, fences, etc.
    - Under munition vulnerability data, a munition is assigned the cookie cutter or Carlton algorithm to evaluate effect.
    - If the munition is to be evaluated using the cookie cutter method then according to the JCATS Algorithm Manual page 3-4:
        - The lethal area is converted to a circular radius
        - If the target is within that radius it is hit; if outside the radius it is not affected.
        - If the target is hit, a roll of the dice determines the effect – KK, MOP, FP, etc.
    - If the munition is to be evaluated using the Carlton method then according to the JCATS Algorithm Manual page 3-4:
        - There is no bounding radius
        - A dice roll against the Carlton probability function determines if the target is hit.
        - If the target is hit, another roll of the dice determines the effect.
        - Any affected entity is also suppressed using "collateral effects".

If the cookie cutter algorithm is used, whether a target is hit or not will be the same through all runs, but the type of kill will be determined by the roll of the dice. With the Carlton algorithm, whether a target is hit or not and the type of kill will both vary from run to run because both are determined by a roll of the dice.

APPENDIX D

DESCRIPTION OF THE JCATS VERIFICATION TEST VIGNETTES AND A SUMMARY OF THE TEST RESULTS

Institute for Defense Analyses

Scenario ID	Scenario Group	Scenario Test	Setup Notes	Operator Notes	Results	Problems	Failed
VV01	LOS and LOF Auto Direct Fire Soldier	target posture: seer standing, target in 7 postures	Use M16 rifle. Seven pairs of seer/target separated by thin buildings.	Set assume enemy ON for all Direct Fire Missions	PH varies with target posture & movement.	#19- LOS in Z-direction not available	
VV02	LOS and LOF Auto Direct Fire Soldier	target posture: seer crouching, target in 7 postures	Use M16 rifle. Seven pairs of seer/target seperated by thin buildings.		PH varies with target posture & movement but not with shooter posture.		
VV03	LOS and LOF Auto Direct Fire Soldier	target posture: seer in pop-up, target in 7 postures	Use M16 rifle. Seven pairs of seer/target separated by thin buildings.		PH does not vary with shooter in pop-up.		
VV04	LOS and LOF Auto Direct Fire Soldier	defilade: seer standing, target in partial or full defilade	Use M16 rifle. Two pairs of seer/target separated by thin buildings.		PH varies with target defilade state.	#5- Question on defilade state reported	
VV05	LOS	full blockage: seer & target standing, 4 types of blockage	Use M16 rifle. Four pairs of seer/target separated by thin buildings.		LOS can be blocked by terrain, high vegetation wiith PLOSB=1, fence ad external wall.		
VV06	LOS	multiple partial blockage: seer & target standing, 4 types of blockage in 6 combinations	Use M16 rifle. Six pairs of seer/target separated by thin buildings.	Do partial in tree steps: (1) 1 fence, (2) 2 fences, (3) other blocking elements	PH varies as height of fence varies.		
VV06b	LOS	multiple partial blockage: seer & target standing, 4 types of blockage in 6 combinations	Use M16 rifle. Six pairs of seer/target separated by thin buildings.		PH changes when second fence is added.		
VV06c	LOS	multiple partial blockage: seer & target standing, 4 types of blockage in 6 combinations	Use M16 rifle. Six pairs of seer/target separated by thin buildings.		PH changes with other blocking elements. However, the effect of vegetation is to attenuate the LOS not to reduce the exposure. If the vegetation is opaque but does not entirely block the target, the vegetation has no effect. If it does block the target, then the target is not acquired.	#1- Error in PH value reported on datevent file when LOS is blocked by vegetation, fence or building. #33- LOS is handled differently for vegetation vs. fences but the results are not consistent and the algorithm description seems to be incomplete.	

Scenario ID	Scenario Group	Scenario Test	Setup Notes	Operator Notes	Results	Problems	Failed
VV07	LOS	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use M16 rifle. Five pairs of seer/target separated by thin buildings.		PH varies by amount of target exposure through window. If target's head is above window, it is not acquired and therefore no shot at.	#4- Error in PH value reported on datevent file when target exposure is reduced by size of window through which target is seen. Also target with head above window is acquired contrary to the algorithm.	
VV08	LOS	full target in window is partially blocked by objects: seer & target standing, 10 combinations of 4 types of partial blockage	Use M16 rifle. Ten pairs of seer/target separated by thin buildings.		Terrain, fences and walls provide partial blockage, but vegetation does not.		
VV09	LOS	partial target in window is blocked by objects: seer & target standing, 5 combinations of 4 types of partial blockage	Use M16 rifle. Five pairs of seer/target separated by thin buildings.	Use window 1.5m high with 1m offset.	The change in PH in this case is from terrain, fences and walls, but not vegetation.		
VV09a	LOS	partial target in window is blocked by objects: seer & target standing, 5 combinations of 4 types of partial blockage	Use M16 rifle. Five pairs of seer/target separated by thin buildings.	Use window 1m high with 1.5m offset.	The change in PH in this case is from reduced exposure of target through window.		
VV10a	LOF - Auto Direct Fire Soldiers	full blockage: seer & target standing, 6 types of blockage	Use M16 rifle. Six pairs of seer/target separated by thin buildings.	Use opaque objects	No LOS means no LOF.	#7- Question on how PLOSB affects kills.	
VV10b	LOF - Auto Direct Fire Soldiers	full blockage: seer & target standing, 6 types of blockage	Use M16 rifle. Six pairs of seer/target separated by thin buildings.	Use clear objects	With clear objects we get LOS and therefore LOF.		

Scenario ID	Scenario Group	Scenario Test	Setup Notes	Operator Notes	Results	Problems	Failed
VV10c	LOF - Auto Direct Fire Soldiers	full blockage: seer & target standing, 6 types of blockage	Use M16 rifle. Six pairs of seer/target separated by thin buildings.	Special tests inside building comparing auto DF and PDF at area.	Auto direct fire can penetrate any clear objects: external wall, interior door or any number of interior walls. Planned direct fire (PDF) at an area does not penetrate external wall, interior door or third interior wall whether they are clear or opaque.	#20- Inconsistence in how fire missions work with clear objects. Normally use opaque objects.	
VV11	LOF - Auto Direct Fire Soldiers	multiple blockage: seer & target standing, 6 types of blockage in 6 combinations	Use M16 rifle. Six pairs of seer/target separated by thin buildings.		No LOS means no LOF.		
VV12	LOF - Auto Direct Fire Soldiers	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use M16 rifle. Five pairs of seer/target separated by thin buildings.		LOS means LOF for target in window. However, if the head of the target is not visible in the window the target will not be acquired and will not be shot at.		
VV13	LOF - Auto Direct Fire Soldiers	flight through window is blocked: seer & target standing, 6 types of blockage	Use M16 rifle. Six pairs of seer/target separated by thin buildings.		No LOS means no LOF.		
VV14	LOF - Auto Direct Fire Soldiers	fight through floors & ceilings: seer & target standing, 2 cases	Use M16 rifle. Two pairs of seer/target in separate buildings.		Cannot perform this test because do not have LOS between floors and therefore will not shoot.		NT
VV15	LOF - PDF Soldier at Soldier	target posture: seer standing, target in 7 postures	Use M16 rifle. Seven pairs of seer/target separated by thin buildings.	Must plan DF mission at target AFTER the shooter acquires the target. Use 1 scond duration of fire and wait for 30 seconds.	PH varies with target posture & movement.	#2- Cannot plan direct fire at a target until shooter acquires target.	
VV16	LOF - PDF Soldier at Soldier	target posture: seer crouching, target in 7 postures	Use M16 rifle. Seven pairs of seer/target separated by thin buildings.	Must plan DF mission at target AFTER the shooter acquires the target. Use 1 scond duration of fire and wait for 30 seconds.	PH varies with target posture & movement but not with shooter posture.		

Scenario ID	Scenario Group	Scenario Test	Setup Notes	Operator Notes	Results	Problems	Failed
VV17	LOF - PDF Soldier at Soldier	target posture: seer in pop-up, target in 7 postures	Use M16 rifle. Seven pairs of seer/target separated by thin buildings.	Must plan DF mission at target AFTER the shooter acquires the target. Use 1 scond duration of fire and wait for 30 seconds.	PH does not vary with shooter in pop-up.	#3- Problem with shooter in pop-up acquiring target . Fixed for Version 3.0	
VV18	LOF - PDF Soldier at Soldier	defilade: seer standing, target in partial or full defilade	Use M16 rifle. Two pairs of seer/target separated by thin buildings.	Must plan DF mission at target AFTER the shooter acquires the target. Use 1 scond duration of fire and wait for 30 seconds.	PH varies with target defilade state.		
VV19	LOF - PDF Soldier at Soldier	full blockage: seer & target standing, 6 types of blockage	Use M16 rifle. Six pairs of seer/target separated by thin buildings.	Must plan DF mission at target AFTER the shooter acquires the target. Use 1 scond duration of fire and wait for 30 seconds.	LOS can be blocked by terrain, high vegetation wiith PLOSB=1, fence ad external wall.		
VV20	LOF - PDF Soldier at Soldier	multiple blockage: seer & target standing, 5 types of blockage in 6 combinations	Use M16 rifle. Five pairs of seer/target separated by thin buildings.	Must plan DF mission at target AFTER the shooter acquires the target. Use 1 scond duration of fire and wait for 30 seconds.	PH changes with other blocking elements. However, the effect of vegetation is to attenuate the LOS not to reduce the exposure. If the vegetation is opaque but does not entirely block the target, the vegetation has not effect. If it does block the target, then the target is not acquired.		
VV21	LOF - PDF Soldier at Soldier	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use M16 rifle. Five pairs of seer/target separated by thin buildings.	Must plan DF mission at target AFTER the shooter acquires the target. Use 1 scond duration of fire and wait for 30 seconds.	PH varies by amount of target exposure through window. If target's head is above window, it is not acquired and therefore no shot at.		

Scenario ID	Scenario Group	Scenario Test	Setup Notes	Operator Notes	Results	Problems	Failed
VV22	LOF - PDF Soldier at Soldier	flight through window is blocked: seer & target standing, 6 types of blockage	Use M16 rifle. Six pairs of seer/target separated by thin buildings.	Must plan DF mission at target AFTER the shooter acquires the target. Use 1 scond duration of fire and wait for 30 seconds.	LOS means LOF for target in window. However, if the head of the target is not visible in the window the target will not be acquired and will not be shot at.		
VV23	LOF - PDF Soldier at Soldier	flight through floors & ceilings: seer & target standing, 2 cases	Use M16 rifle. Two pairs of seer/target in separate floors.	Must plan DF mission at target AFTER the shooter acquires the target. Use 1 scond duration of fire and wait for 30 seconds.	Cannot perform this test because do not have LOS between floors and therefore will not shoot.		NT
VV24	LOF - PDF at Area with Soldiers	target posture: seer standing, target in 7 postures	Use M16 automatic rifle. Seven pairs of seer/target separated by thin buildings.	Wait for 30 seconds and shoot for 10 seconds.	No effect of target posture or movement.	#11- Direct Fire attacks produce records in the datevent file that do not match the description.	
VV25	LOF - PDF at Area with Soldiers	target posture: seer crouching, target in 7 postures	Use M16 automatic rifle. Seven pairs of seer/target separated by thin buildings.	Wait for 30 seconds and shoot for 10 seconds.	No effect of target posture or movement. No effect of shooter posture.		
VV26	LOF - PDF at Area with Soldiers	target posture: seer in pop-up, target in 7 postures	Use M16 automatic rifle. Seven pairs of seer/target separated by thin buildings.	Wait for 30 seconds and shoot for 10 seconds.	No effect of shooter in pop up.	#3- Problem with shooter in pop-up acquiring target . Fixed for Version 3.0	
VV27	LOF - PDF at Area with Soldiers	defilade: seer standing, target in partial or full defilade	Use M16 automatic rifle. Two pairs of seer/target separated by thin buildings.	Wait for 30 seconds and shoot for 10 seconds.	No effect of target defilade state.		
VV28	LOF - PDF at Area with Soldiers	full blockage: seer & target standing, 6 types of blockage	Use M16 automatic rifle. Six pairs of seer/target separated by thin buildings.	Wait for 30 seconds and shoot for 10 seconds.	Terrain, fences, external walls and third interior wall block LOF, but vegetation does not. First and secon interior wall do not block LOF.		

Scenario ID	Scenario Group	Scenario Test	Setup Notes	Operator Notes	Results	Problems	Failed
VV29	LOF - PDF at Area with Soldiers	multiple blockage: seer & target standing, 5 types of blockage in 6 combinations	Use M16 automatic rifle. Five pairs of seer/target separated by thin buildings.	Wait for 30 seconds and shoot for 10 seconds.	Terrain, fences, external walls and third interior wall block LOF, but vegetation does not. First and secon interior wall do not block LOF.		
VV30	LOF - PDF at Area with Soldiers	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use M16 automatic rifle. Five pairs of seer/target separated by thin buildings.	Wait for 30 seconds and shoot for 10 seconds.	PDF at area fires through window. Model does not check for a target so the head above the window has no effect.		
VV31	LOF - PDF at Area with Soldiers	flight through window is blocked: seer & target standing, 6 types of blockage	Use M16 automatic rifle. Six pairs of seer/target separated by thin buildings.	Wait for 30 seconds and shoot for 10 seconds.	Terrain, fences, external walls and third interior wall block LOF, but vegetation does not. First and second interior wall do not block LOF.		
VV32	LOF - PDF at Area with Soldiers	fight through floors & ceilings: seer & target standing, 2 cases	Use M16 automatic rifle. Two pairs of seer/target in separate buildings.	Wait for 30 seconds and shoot for 10 seconds.	Shooter fires at area but LOF is blocked by floors and ceilings.	#15- Need to document how to set up firing between floors.	
VV33	LOF - Planned Indirect Fire	target posture: seer standing, target in 7 postures	Use M79 grenade launcher automatic rifle. Seven pairs of seer/target separated by thin buildings.	Shoot OFF for all entities. Make sure shooter has indirect fire capability. Set up artillery fire mission.	No effect of target posture or movement. No effect of shooter posture.	#28- For indirect fire, aim point not impact point is being reported on the IA record in the datevent file.	
VV34	LOF - Planned Indirect Fire	defilade: seer standing, target in partial or full defilade	Use MLRS. Two pairs of seer/target separated by thin buildings.	Shoot OFF for all entities. Make sure shooter has indirect fire capability. Set up artillery fire mission.	No effect of target defilade state.		
VV35	LOF - Planned Indirect Fire	full blockage: seer & target standing, 6 types of blockage	Use MLRS. Six pairs of seer/target separated by thin buildings.	Shoot OFF for all entities. Make sure shooter has indirect fire capability. Set up artillery fire mission.	Terrain, fences, external walls block LOF, but vegetation does not. Could not test inside buildings.	#20- Inconsistence in how fire missions work with clear objects. Normally use opaque objects. #30- Cannot fire indirect fire mission at target line that is inside a building.	X

Scenario ID	Scenario Group	Scenario Test	Setup Notes	Operator Notes	Results	Problems	Failed
VV36	LOF - Planned Indirect Fire	multiple blockage: seer & target standing, 5 types of blockage in 6 combinations	Use MLRS. Five pairs of seer/target separated by thin buildings.	Shoot OFF for all entities. Make sure shooter has indirect fire capability. Set up artillery fire mission.	Terrain, fences, external walls block LOF, but vegetation does not. Could not test inside buildings.		
VV37	LOF - Planned Indirect Fire	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use M79 grenade launcher automatic rifle. Five pairs of seer/target separated by thin buildings.	Shoot OFF for all entities. Make sure shooter has indirect fire capability. Set up artillery fire mission.	PIF fires at area through window. Model does not check for a target so the head above the window has no effect.		
VV38	LOF - Planned Indirect Fire	flight through window is blocked: seer & target standing, 6 types of blockage	Use MLRS. Six pairs of seer/target separated by thin buildings.	Shoot OFF for all entities. Make sure shooter has indirect fire capability. Set up artillery fire mission.	Terrain, fences, external walls block LOF, but vegetation does not. Could not test inside buildings.	#30- Cannot fire indirect fire mission at target line that is inside a building.	X
VV39	LOF - Planned Indirect Fire	fight through floors & ceilings: seer & target standing, 2 cases	Use M79 grenade launcher automatic rifle. Two pairs of seer/target in separate buildings.	Shoot OFF for all entities. Make sure shooter has indirect fire capability. Set up artillery fire mission.	Cannot perform this test because PIF not performed inside buildings.		NT
VV40	LOF - Auto Indirect Fire with FO	target posture: seer standing, target in 7 postures	Use 120 mm with regular ammo automatic rifle. Seven pairs of seer/target separated by thin buildings.		No effect of target posture or movement. No effect of shooter posture.		
VV41	LOF - Auto Indirect Fire with FO	defilade: seer standing, target in partial or full defilade	Use 120 mm with regular ammo automatic rifle. Two pairs of seer/target separated by thin buildings.		No effect of target defilade state.		
VV42	LOF - Auto Indirect Fire with FO	full blockage: seer & target standing, 6 types of blockage	Use MLRS. Six pairs of seer/target separated by thin buildings.		Terrain, fences, external walls block LOF, but vegetation does not. Could not test inside buildings.	#30- Cannot fire indirect fire mission at target line that is inside a building.	X

Scenario ID	Scenario Group	Scenario Test	Setup Notes	Operator Notes	Results	Problems	Failed
VV43	LOF - Auto Indirect Fire with FO	multiple blockage: seer & target standing, 5 types of blockage in 6 combinations	Use MLRS. Six pairs of seer/target separated by thin buildings.		Terrain, fences, external walls block LOF, but vegetation does not. Could not test inside buildings.	#30- Cannot fire indirect fire mission at target line that is inside a building.	X
VV44	LOF - Auto Indirect Fire with FO	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use 120 mm with regular ammo automatic rifle. Five pairs of seer/target separated by thin buildings.		Indirect Fire with FO fires at area through window. Forward observer cannot acquire a target with the head above the window so shooter does not fire at target.		
VV45	LOF - Auto Indirect Fire with FO	flight through window is blocked: seer & target standing, 6 types of blockage	Use MLRS. Six pairs of seer/target separated by thin buildings.		Terrain, fences, external walls block LOF, but vegetation does not. Could not test inside buildings.	#30- Cannot fire indirect fire mission at target line that is inside a building.	X
VV46	LOF - Auto Indirect Fire with FO	flight through floors & ceilings: seer & target standing, 2 cases	Use 120 mm with regular ammo automatic rifle. Two pairs of seer/target in separate buildings.		Cannot perform this test.		NT
VV47	LOF - Direct Fire with LD	target posture: seer standing, target in 7 postures	Use 120 mm with regular ammo automatic rifle. Seven pairs of seer/target separated by thin buildings.		No effect of target posture or movement. No effect of shooter posture.	#10- Direct Support Fire with Laser Designator is reported as Direct Fire records (SD,ID,ED) in the datevent file, contrary to JCATS documentation which says it is reported as Indirect Fire. #14-	
VV48	LOF - Direct Fire with LD	defilade: seer standing, target in partial or full defilade	Use 120 mm with regular ammo automatic rifle. Two pairs of seer/target separated by thin buildings.		No effect of target defilade state.		

Scenario ID	Scenario Group	Scenario Test	Setup Notes	Operator Notes	Results	Problems	Failed
VV49	LOF - Direct Fire with LD	full blockage: seer & target standing, 6 types of blockage	Use 120 mm with regular ammo automatic rifle. Six pairs of seer/target separated by thin buildings.			#20- Inconsistence in how fire missions work with clear objects. Normally use opaque objects. #34-Cannot get building to block fire.	X
VV50	LOF - Direct Fire with LD	multiple blockage: seer & target standing, 5 types of blockage in 6 combinations	Use 120 mm with regular ammo automatic rifle. Five pairs of seer/target separated by thin buildings.			#34- Cannot get building to block fire.	X
VV51	LOF - Direct Fire with LD	target visible through window: seer standing outside & target in two postures inside with 5 visibility situations	Use 120 mm with regular ammo automatic rifle. Five pairs of seer/target separated by thin buildings.	LD outside and target inside.	LD does not acquire target with head above the window and so shooter does not fire. Other targets are acquired and fired on.		
VV52	LOF - Direct Fire with LD	flight through window is blocked: seer & target standing, 6 types of blockage	Use 120 mm with regular ammo automatic rifle. Six pairs of seer/target separated by thin buildings.			#34- Cannot get building to block fire.	X
VV53	LOF - Direct Fire with LD	flight through floors & ceilings: seer & target standing, 2 cases	Use 120 mm with regular ammo automatic rifle. Two pairs of seer/target in separate buildings.		With LD and target both on first floor shooter fires at target, through a 5 story building.	How does shooter pick up laser inside building? #34- DS with LD does not work properly with/inside of buildings.	X
VV54	Soldier Movement	posture & terrain with no micro terrain; 3 postures, 3 terrain inclines	Use 9 soldiers		Speed of soldier is affected by steepness of the terrain.	#9- Reported speed of rifleman over elevated terrain is not consistent.	
VV55	Soldier Movement	posture & terrain on road; 3 postures, 3 terrain inclines	Use 9 soldiers		Speed of soldier is affected by steepness of the terrain.		
VV56	Soldier Movement	posture & terrain on grass; 3 postures, 3 terrain inclines	Use 9 soldiers		Speed of soldier is affected by steepness of the terrain.		

Scenario ID	Scenario Group	Scenario Test	Setup Notes	Operator Notes	Results	Problems	Failed
VV57	Soldier Movement	posture & terrain on other vegetation; 3 postures, 3 type vegetation, flat ground	Use 14 soldiers			#29- Trafficability factor is not used in determining speed over other types of vegetation.	
VV58	Soldier Movement	blocking, breaching & penetration: soldier walking on road, 6 types of blockers in 14 cases	Use 14 soldiers. Use engineering object with B=0 P=0 for no B or P, otherwise use object with both B and P capability and turn breach on to breach and off to penetrate.	Solid 3.5 fence B=0, P=0. Solid 3m fence b=26 p=28. Wall type 2 B=0 P=0. Wall type 1 B=9 P=2. Door type 2 B=0 P=0. Door type 1 B=1 P=2. Note must put door type 2 in wall type 2 to get stoppage, otherwise code breaches because wall can be breached.	Soldiers breach and penetrate according to the rules. Vegetation does not block movement. Vehicles do not block people. Soldiers cannot breach building shells.	#13- Additions needed for breach and penetration. #25- Second soldier through breach may not be delayed until breach has been completed by first soldier.	X
VV59	Soldier Movement	movement in buildings: soldier walking inside building 2 blocking entities, breach and penetrate	Use 6 soldiers.	See notes under #58 about objects. Ramp must connect to building in order for soldier to enter building.	Soldiers can move in buildings and can breach and penetrate interior doors and walls.		
VV60	Soldier Movement	entering building: (1) 3 postures, soldier enters through exterior window, (2) 2 postures, soldier entering via one story ramp, (3) 3 postures, soldier entering via 3 story ramp	Use 8 soldiers.	Scenario #59 and #60 were performed under one test vv_59.	Speed of soldier changes with incline of ramp. Soldiers can enter buildings from ramp or through exterior window.	#8- In version 2.3 ramps were treated as elevated highways. This was fixed in version 3.0.	
VV61	Soldier Movement	rubble: 3 postures, 3 terrain inclines.	Use 9 soldiers.		Rubble slows down soldier movement to the maximum speed for rubble.	#16- Rubble appears only on the backside of a building when it is hit in the front.	
VV62	Blocking	blocking in column: 3 tanks in 6 cases of movement	Use 6 columns of 3 tanks each		Tanks block and delay movement of other tanks until the way is cleared.		

Scenario ID	Scenario Group	Scenario Test	Setup Notes	Operator Notes	Results	Problems	Failed
VV63	Blocking	Other Blocking: 2 tanks, 2 soldiers, or a tank and a soldier	8 cases of blocking		Tanks block other tanks. Two tanks blocking each other can move away at 90 degrees. People and tanks do not block each other.	#12- The angle at which a tank can move away from another tank that has blocked it is not consistent.	
VV64	Miscellaneou s	Bullet proof workaround option 1			For Direct Fire get LOS and therefore get LOF. For Indirect Fire bullet will be stopped. Not a good workaround.	Does not work as a workaround. #20- Inconsistence in how fire missions work with clear objects. Normally use opaque objects.	
VV65	Miscellaneou s	Bullet proof workaround option 2			If three walls are see-thru, for Direct Fire get LOS and therefore acquire target and automatically get LOF because had LOS. If three walls are not see-thru, do not get LOS to target and therefore for Direct Fire dc not shoot. For Indirect Fire bullet will be stopped in either case. Not a good workaround.	Does not work as a workaround. #20- Inconsistence in how fire missions work with clear objects. Normally use opaque objects.	
VV66	Miscellaneou s	Test LOS and LOF on ramp			Get LOS and can shoot him.	#8- In version 2.3 ramps were treated as elevated highways. This was fixed in version 3.0.	
VV67	Miscellaneou s	Test LOS and LOF on stairs			Get LOS as long as both entities are on same floor. When entity goes to next floor on activity node, no longer get LOS. Can shoot as long as on same floor.		

Scenario ID	Scenario Group	Scenario Test	Setup Notes	Operator Notes	Results	Problems	Failed
VV68	Miscellaneous	Test to see if can penetrate fence and if type of material used for fence matters.			Can penetrate. It does not vary directly by type of material but each type of fence can be assigned a different terrain code that determines the time to penetrate the fence.		
VV69	Miscellaneous	Test stacking of terrain			No change here as soldiers move between different surfaces.	#29- Trafficability factor is not used in determining speed over other types of vegetation.	
VV70	Miscellaneous	Test LOF on fences with different materials.			Fence stops LOF as tested with planned Direct Fire at an area. It does not depend on the material of the fence.		

APPENDIX E

JCATS VERIFICATION TEST RESULTS

Institute for Defense Analyses

VV	ID	Seer Location	Target Location	Seer Posture	Target Posture	Blocking Terrain	Blocking Vegetation	Blocking Fence	Blocking Exterior Wall	Visible through Window	Head Above Window	Comments	Shooter ID	Target ID	Range in meters	Result: PH	Expected PH	Fail / NT=no test
1	1	outside	outside	standing	standing							effect of target posture	11	18	100	0.640	0.640	
	2	outside	outside	standing	crouching								12	19	100	0.639	0.640	
	3	outside	outside	standing	crawling								13	20	100	0.321	0.320	
	4	outside	outside	standing	prone								14	21	100	0.080		
	5	outside	outside	standing	foxhole								15	22	100	0.032		
	6	outside	outside	standing	walking								16	23	100	0.319	0.320	
	7	outside	outside	standing	running								17	24	100	0.320	0.320	
2	8	outside	outside	crouching	standing							effect of target posture	11	18	100	0.641	0.640	
	9	outside	outside	crouching	crouching								12	19	100	0.639	0.640	
	10	outside	outside	crouching	crawling								13	20	100	0.321	0.320	
	11	outside	outside	crouching	prone								14	21	100	0.080		
	12	outside	outside	crouching	foxhole								15	22	100	0.032		
	13	outside	outside	crouching	walking								16	23	100	0.319	0.320	
	14	outside	outside	crouching	running								17	24	100	0.319	0.320	
3	15	outside	outside	pop-up	standing							effect of target posture on seer pop-up	11	18	100	0.639	0.640	
	16	outside	outside	pop-up	crouching								12	19	100	0.640	0.640	
	17	outside	outside	pop-up	crawling								13	20	100	0.320	0.320	
	18	outside	outside	pop-up	prone								14	21	100	0.080		
	19	outside	outside	pop-up	foxhole								15	22	100	0.032		
	20	outside	outside	pop-up	walking								16	23	100	0.320	0.320	
	21	outside	outside	pop-up	running								17	24	100	0.320	0.320	
4	22	outside	outside	standing	standing - partial defilade							effect of defilade - use engineering object to create defilade state	14	21	100	0.160		
	23	outside	outside	standing	prone - full defilade								15	22	100	0.032		
	23b	outside	outside	standing	open - standing								11	18	100	0.640		
5	24	outside	outside	standing	standing	full						effect of full blockage	16			no acquire		
	25	outside	outside	standing	standing		full						15			no acquire		
	26	outside	outside	standing	standing			full					11			no acquire		
	27	outside	outside	standing	standing				full				14			no acquire		
6	28	outside	outside	standing	standing			.5m				effects of multiple blockages	15	24	59	0.779		
	29	outside	outside	standing	standing			1m				start with one fence at 34m from shooter	11	20	59	0.781		
	30	outside	outside	standing	standing			1.25m					12	23	59	0.777		
	31	outside	outside	standing	standing			1.5m					13	18	59	0.197		
	32	outside	outside	standing	standing			1.6m					14	21	59	no acquire		
	33	outside	outside	standing	standing			1.7m					17	19	59	no acquire		
6b	28	outside	outside	standing	standing			.5m				effects of multiple blockages	15	24	60	0.112		
	29	outside	outside	standing	standing			1m				put one fence at 34m from shooter and a second fence of 1.5m just in front of the target which is 60 m from the shooter	11	20	60	0.112		
	30	outside	outside	standing	standing			1.25m					12	23	60	0.112		
	31	outside	outside	standing	standing			1.5m					13	18	60	0.112		
	32	outside	outside	standing	standing			1.6m					14	21	60	no acquire		
	33	outside	outside	standing	standing			1.7m					17	19	60	no acquire		
6c	28	outside	outside	standing	standing	partial	partial					effects of multiple blockages	15	24	60	0.780		
	29	outside	outside	standing	standing	partial		partial				finally perform tests with different types of blocking entities	11	20	60	0.115		

VV	ID	Seer Location	Target Location	Seer Posture	Target Posture	Blocking Terrain	Blocking Vegetation	Blocking Fence	Blocking Exterior Wall	Visible through Window	Head Above Window	Comments	Shooter ID	Target ID	Range in meters	Result: PH	Expected PH	Fail / NT=no test
	30	outside	outside	standing	standing	partial			partial				12	23	60	0.115		
	31	outside	outside	standing	standing		partial	partial					13	18	60	0.116		
	32	outside	outside	standing	standing		partial		partial				14	21	60	0.112		
	33	outside	outside	standing	standing			partial	partial				17	19	60	0.112		
7	34	outside	inside	standing	standing					full		target visible through window	15	24	20	0.920		
	35	outside	inside	standing	standing					partial			11	20	20	0.264		
	36	outside	inside	standing	standing					partial	yes		12	23	20	no acquire		
	37	outside	inside	standing	crouching					full			13	18	20	0.920		
	38	outside	inside	standing	crouching					partial			14	21	20	0.178		
8	39	outside	inside	standing	standing	partial				full		target in window blocked	30		60	0.779		
	40	outside	inside	standing	standing		partial			full			8		60	0.781		
	41	outside	inside	standing	standing			partial		full			28		62	0.145		
	42	outside	inside	standing	standing				partial	full			29		61	0.124		
	43	outside	inside	standing	standing	partial	partial			full			7		60	0.781		
	44	outside	inside	standing	standing	partial		partial		full			9		58	0.126		
	45	outside	inside	standing	standing	partial			partial	full			10		61	0.124		
	46	outside	inside	standing	standing		partial	partial		full			25		59	0.123		
	47	outside	inside	standing	standing		partial		partial	full			26		61	0.128		
	48	outside	inside	standing	standing			partial	partial	full			27		61	0.145		
9	49	outside	inside	standing	standing	partial				partial		partial target in window blocked	30		60	0.779		
	50	outside	inside	standing	standing		partial	partial		partial		window 1.5m high with 1m offset	25		59	0.123		
	51	outside	inside	standing	standing			partial	partial	partial			27		61	0.145		
	52	outside	inside	standing	standing		partial		partial	partial	yes		26		61	no acquire		
	53	outside	inside	standing	standing	partial		partial		partial	yes		9		61	no acquire		
9a	49	outside	inside	standing	standing	partial				partial		partial target in window blocked	30		60	0.115		
	50	outside	inside	standing	standing		partial	partial		partial		window 1m high 1.5m offset	25		59	0.116		
	51	outside	inside	standing	standing			partial	partial	partial			27		61	0.115		
	52	outside	inside	standing	standing		partial		partial	partial	yes		26		61	no acquire		
	53	outside	inside	standing	standing	partial		partial		partial	yes		9		61	no acquire		

Note: Use M16 weapon

Note: Set ups for windows in #7, etc.

Height	Offset	Posture	Defilade
3m	0m	standing	full
1m	1.3m	standing	partial
1m	.5m	standing	no head
3m	0m	crouching	full
1m	.5m	crouching	partial

VV	ID	Shooter Location	Target Location	Shooter Posture	Target Posture	Blocking Terrain	Blocking Vegetation	Blocking Fence	Blocking Exterior Wall	2 Interior Walls	3 Interior Walls	Visible through Window	Head Above Window	Comments	Shooter ID	Target ID	Range	Result: PH	Fail / NT=no test	Result: Target Hit
1	1	outside	outside	standing	standing									effect of target posture	11	18	100	0.640		K
	2	outside	outside	standing	crouching										12	19	100	0.639		K
	3	outside	outside	standing	crawling										13	20	100	0.321		K
	4	outside	outside	standing	prone										14	21	100	0.080		K
	5	outside	outside	standing	foxhole										15	22	100	0.032		S
	6	outside	outside	standing	walking										16	23	100	0.319		K
	7	outside	outside	standing	running										17	24	100	0.320		K
2	8	outside	outside	crouching	standing									effect of target posture	11	18	100	0.641		K
	9	outside	outside	crouching	crouching										12	19	100	0.639		K
	10	outside	outside	crouching	crawling										13	20	100	0.321		K
	11	outside	outside	crouching	prone										14	21	100	0.080		K
	12	outside	outside	crouching	foxhole										15	22	100	0.032		
	13	outside	outside	crouching	walking										16	23	100	0.319		K
	14	outside	outside	crouching	running										17	24	100	0.319		K
3	15	outside	outside	pop-up	standing									effect of target posture on seer pop-up	11	18	100	0.639		K
	16	outside	outside	pop-up	crouching										12	19	100	0.640		K
	17	outside	outside	pop-up	crawling										13	20	100	0.320		K
	18	outside	outside	pop-up	prote										14	21	100	0.080		K
	19	outside	outside	pop-up	foxhole										15	22	100	0.032		S
	20	outside	outside	pop-up	walking										16	23	100	0.320		K
	21	outside	outside	pop-up	running										17	24	100	0.320		K
4	22	outside	outside	standing	standing - partial defilade									effect of defilade	14	21	100	0.160		K
	23	outside	outside	standing	prone - full defilade										15	22	100	0.032		
	23b	outside	outside	standing	in open - standing										11	18	100	0.640		K
10a	24	outside	outside	standing	standing	x								effect of single blockage	16			no LOS		
	25	outside	outside	standing	standing		x							when cannot see through objects	15	24		no LOS		
	26	outside	outside	standing	standing			x							11	20		no LOS		
	27	outside	outside	standing	standing				x					**There is no LOS so no firing of weapon**	12	23		no LOS		
	28	inside	inside	standing	standing					x					13	18		no LOS		
	29	inside	inside	standing	standing						x				14	21		no LOS		
10b	24	outside	outside	standing	standing	x								effect of single blockage	16		?	no LOS		cannot make terrain clear
	25	outside	outside	standing	standing		x							when can see through objects	15	24		0.850		K
	26	outside	outside	standing	standing			x							11	20		0.848		K
	27	outside	outside	standing	standing				x						12	23		0.846		K
	28	inside	inside	standing	standing					x					13	18		0.955		K
	29	inside	inside	standing	standing						x				14	21		0.939		K
10c	24	outside	outside	standing	standing	x								effect of single blockage						
	25	outside	outside	standing	standing		x													
	26	outside	outside	standing	standing			x												
	27	outside	outside	standing	standing				x											
	28	inside	inside	standing	standing					x										
	29	inside	inside	standing	standing						x									
11	30	outside	outside	standing	standing	x		x						effects of multiple blockages				no LOS		
	31	outside	outside	standing	standing		x	x										no LOS		
	32	outside	outside	standing	standing			x	x					**There is no LOS so no firing of weapon**				no LOS		
	33	outside	inside	standing	standing			x		x								no LOS		
	34	outside	inside	standing	standing			x			x							no LOS		
12	35	outside	inside	standing	standing							full		target visible through window	15	24	20	0.920		K
	36	outside	inside	standing	standing							partial			11	20	20	0.264		S

VV	ID	Shooter Location	Target Location	Shooter Posture	Target Posture	Blocking Terrain	Blocking Vegetation	Blocking Fence	Blocking Exterior Wall	2 Interior Walls	3 Interior Walls	Visible through Window	Head Above Window	Comments	Shooter ID	Target ID	Range	Result: PH	Fail / NT=no test	Result: Target Hit
	37	outside	inside	standing	standing							partial	yes		12	23	20	no acquire		
	38	outside	inside	standing	crouching							full			13	18	20	0.920		K
	39	outside	inside	standing	crouching							partial			14	21	20	0.180		K
13	40	outside	inside	standing	standing	x						in path		flight through window blocked	16			no LOS		
	41	outside	inside	standing	standing		x					in path			15	24		no LOS		
	42	outside	inside	standing	standing			x				in path		**There is no LOS so no firing of weapon**	11	20		no LOS		
	43	outside	inside	standing	standing				x			in path			12	23		no LOS		
	44	outside	inside	standing	standing					x		in path			13	18		no LOS		
	45	outside	inside	standing	standing						x	in path			14	21		no LOS		
14	46	Floor 1	Floor 2	standing	standing									flight through floors & ceilings				no LOS	NT	
	47	Floor 2	Floor 1	standing	standing									**There is no LOS so no firing of weapon**				no LOS	NT	

Note: These cases are for Auto Direct Fire at entity, namely, people against people using rifle. According to LLNL in this case LOS implies LOF.
Compare these results to LOS results to see if this is the case.

Note: Use M16 weapon

VV	ID	Shooter Location	Target Location	Shooter Posture	Target Posture	Blocking Terrain	Blocking Vegetation	Blocking Fence	Blocking Exterior Wall	2 Interior Walls	3 Interior Walls	Visible through Window	Head Above Window	Comments	Shooter ID	Target ID	Range in Meters	Result: PH	Fail / NT=no test	Result: Target Hit
15	1	outside	outside	standing	standing									effect of target posture	11	18	100	0.640		KK
	2	outside	outside	standing	crouching										12	19	100	0.639		KK
	3	outside	outside	standing	crawling										13	20	100	0.321		KK
	4	outside	outside	standing	prone										14	21	100	0.080		S
	5	outside	outside	standing	foxhole										15	22	100	0.032		
	6	outside	outside	standing	walking										16	23	100	0.319		K
	7	outside	outside	standing	running										17	24	100	0.319		K
16	8	outside	outside	crouching	standing									effect of target posture	11	18	100	0.641		K
	9	outside	outside	crouching	crouching										12	19	100	0.639		K
	10	outside	outside	crouching	crawling										13	20	100	0.321		K
	11	outside	outside	crouching	prone										14	21	100	0.080		S
	12	outside	outside	crouching	foxhole										15	22	100	0.032		K
	13	outside	outside	crouching	walking										16	23	100	0.320		K
	14	outside	outside	crouching	running										17	24	100	0.639		K
17	15	outside	outside	pop-up	standing									effect of target posture on seer pop-up	11	18	100	0.639		K
	16	outside	outside	pop-up	crouching										12	19	100	0.640		K
	17	outside	outside	pop-up	crawling										13	20	100	0.320		K
	18	outside	outside	pop-up	prone										14	21	100	0.080		S
	19	outside	outside	pop-up	foxhole										15	22	100	0.032		K
	20	outside	outside	pop-up	walking										16	23	100	0.319		K
	21	outside	outside	pop-up	running										17	24	100	0.319		K
18	22	outside	outside	standing	standing - partial defilade									effect of defilade	14	21	100	0.160		K
	23	outside	outside	standing	prone- full										15	22	100	0.032		MISS
	23b	outside	outside	standing	Stand - open										11	18	100	0.640		K
19	24	outside	outside	standing	standing	x								effect of single blockage				no LOS		
	25	outside	outside	standing	standing		x											no LOS		
	26	outside	outside	standing	standing			x						**There is no LOS so no firing of weapon**				no LOS		
	27	outside	outside	standing	standing				x									no LOS		
	28	inside	inside	standing	standing					x								no LOS		
	29	inside	inside	standing	standing						x							no LOS		
20	30	outside	outside	standing	standing	x		x						effects of multiple blockages				no LOS		
	31	outside	outside	standing	standing		x	x										no LOS		
	32	outside	outside	standing	standing			x	x					**There is no LOS so no firing of weapon**				no LOS		
	33	inside	inside	standing	standing			x		x								no LOS		
	34	inside	inside	standing	standing			x			x							no LOS		
21	35	outside	inside	standing	standing							full		target visible through window	15	24	20	0.920		K
	36	outside	inside	standing	standing							partial			11	20	20	0.264		K
	37	outside	inside	standing	standing							partial	yes		12		20	no acq		
	38	outside	inside	standing	crouching							full			13	18	20	0.920		K
	39	outside	inside	standing	crouching							partial			14	21	20	0.178		K
22	40	outside	inside	standing	standing	x						in path		flight through window blocked				no LOS		
	41	outside	inside	standing	standing		x					in path						no LOS		
	42	outside	inside	standing	standing			x				in path		**There is no LOS so no firing of weapon**				no LOS		
	43	outside	inside	standing	standing				x			in path						no LOS		
	44	outside	inside	standing	standing					x		in path						no LOS		
	45	outside	inside	standing	standing						x	in path						no LOS		
23	46	Floor 1	Floor 2	standing	standing									flight through floors & ceilings				no LOS	NT	
	47	Floor 2	Floor 1	standing	standing									**There is no LOS so no firing of weapon**				no LOS	NT	
	Note: These cases are for Planned Direct Fire at entity, namely, people against people using rifle. According to LLNL in this case LOS implies LOF.																			
	Compare these results to LOS results to see if this is the case.																			
	This type of mission cannot be planned until the target has been acquired. Therefore, the mission must be set up by the tester during the simulation.																			
	Note: Use M16 weapon																			

VV	ID	Shooter Location	Target Location	Shooter Posture	Target Posture	Blocking Terrain	Blocking Vegetation	Blocking Fence	Blocking Exterior Wall	2 Interior Walls	3 Interior Walls	Visible through Window	Head Above Window	Comments	Shooter ID	Target ID	Fire	Block	Result: Target Hit	Fail / NT=no test
24	1	outside	outside	standing	standing									effect of target posture	11	18	x			
	2	outside	outside	standing	crouching									No effect of posture.	12	19	x		K	
	3	outside	outside	standing	crawling										13	20	x		K	
	4	outside	outside	standing	prone										14	21	x		K	
	5	outside	outside	standing	foxhole										15	22	x		K	
	6	outside	outside	standing	walking										16	23	x			
	7	outside	outside	standing	running										17	24	x			
25	8	outside	outside	crouching	standing									effect of target posture	11	18	x			
	9	outside	outside	crouching	crouching									No effect of posture.	12	19	x		K	
	10	outside	outside	crouching	crawling										13	20	x		S	
	11	outside	outside	crouching	prone										14	21	x		K	
	12	outside	outside	crouching	foxhole										15	22	x		K	
	13	outside	outside	crouching	walking										16	23	x		S	
	14	outside	outside	crouching	running										17	24	x			
26	15	outside	outside	pop-up	standing									effect of target posture on seer pop-up	11	18	x		K	
	16	outside	outside	pop-up	crouching									No effect of posture.	12	19	x			
	17	outside	outside	pop-up	crawling										13	20	x			
	18	outside	outside	pop-up	prone										14	21	x		K	
	19	outside	outside	pop-up	foxhole										15	22	x		K	
	20	outside	outside	pop-up	walking										16	23	x			
	21	outside	outside	pop-up	running										17	24	x		S	
27	22	outside	outside	standing	standing - partial defilade									effect of defilade	14	21	x		K	
	23	outside	outside	standing	prone - full defilade									No effect of defilade.	15	22	x			
	23b	outside	outside	standing	stand - open										11	18	x		K	
28	24	outside	outside	standing	standing	x								effect of single blockage	16		x	B		
	25	outside	outside	standing	standing		x								15	24	x	no block vegetation	K	
	26	outside	outside	standing	standing			x							11		x	B		
	27	outside	outside	standing	standing				x						12		x	B		
	28	inside	inside	standing	standing					x					13	18	x	no	K	
	29	inside	inside	standing	standing						x				14		x	B 3rd wall		
29	30	outside	outside	standing	standing	x		x						effects of multiple blockages	16		x	B		
	31	outside	outside	standing	standing		x	x						1.5m Fence first	15	24	x	B		
	32	outside	outside	standing	standing			x	x						12		x	B		
	33	inside	inside	standing	standing			x		x					13	18	x	B		
	34	inside	inside	standing	standing			x			x				14		x	B		
30	35	outside	inside	standing	standing							full		target visible through window	15	24	x		K	
	36	outside	inside	standing	standing							partial		Bullet goes through window to 3rd wall	11	20	x		K	
	37	outside	inside	standing	standing							partial	yes		12	23	x		K	
	38	outside	inside	standing	crouching							full			13		x			
	39	outside	inside	standing	crouching							partial			14	21	x		S	
31	40	outside	inside	standing	standing	x						in path		flight through window blocked	16		x	B		
	41	outside	inside	standing	standing		x					in path			15		x	no block vegetation	K	
	42	outside	inside	standing	standing			x				in path			11		x	B		
	43	outside	inside	standing	standing				x			in path			12		x	B		
	44	outside	inside	standing	standing					x		in path			13		x	no	?	
	45	outside	inside	standing	standing						x	in path			14		x	B 3rd wall		
32	46	Floor 1	Floor 2	standing	standing									flight through floors & ceilings	13		x	B		
	47	Floor 2	Floor 1	standing	standing										14		x	B		

Note: These cases are for Planned Direct Fire at an area, namely people shoot rifle at an area in which there are people targets.
According to LLNL this type of fire should use the LOF algorithms. Not sure how to test for this. Intended target creates ED record, unintended target creates EA record.

Note: Use M16 weapon on automatic

VV	ID	Shooter Location	Target Location	Target Posture	Blocking Terrain	Blocking Vegetation	Blocking Fence	Blocking Exterior Wall	2 Interior Walls	3 Interior Walls	Visible through Window	Head Above Window	Comments	Shooter ID	Target ID	Fire	Block	Result: Target Hit	Note	Fail / NT=no test
33	1	outside	outside	standing									effect of target posture	11	18	x		S		
	2	outside	outside	crouching									No effect of posture.	12	19	x		S		
	3	outside	outside	crawling										13	20	x		S		
	4	outside	outside	prone										14	21	x		S		
	5	outside	outside	foxhole										15	22	x		S		
	6	outside	outside	walking										16	23	x		S		
	7	outside	outside	running										17	24	x		S		
34	8	outside	outside	standing - partial defilade									effect of defilade	11	24	x		S		
	9	outside	outside	prone - full defilade									No effect of defilade.	16	22	x		S		
	9b	outside	outside	in open - standing										17	19	x		S		
35	10	outside	outside	standing	x								effect of single blockage	6			B			
	11	outside	outside	standing		x								12			no	S	vegetation does not block	
	12	outside	outside	standing			x							10			B			
	13	outside	outside	standing				x						7			B			
	14	outside	inside	standing					x					11				**Problem with interior of building**		F
	15	outside	inside	standing						x				8				**Problem with interior of building**		F
36	16	outside	outside	standing	x		x						effects of multiple blockages	6			B			
	17	outside	outside	standing		x	x						1.5M fence in front always blocks	12			B			
	18	outside	outside	standing			x	x						7			B			
	19	outside	inside	standing			x		x					11			B			
	20	outside	inside	standing			x			x				8			B			
37	21	outside	inside	standing							full		target visible through window	15		x		S		
	22	outside	inside	standing							partial			11		x		S		
	23	outside	inside	standing							partial	yes		12		x		S	Target is exposed so head above window has no effect.	
	24	outside	inside	crouching							full			13		x		S		
	25	outside	inside	crouching							partial			14		x		S		
38	26	outside	inside	standing	x						in path		flight through window blocked	6			B			
	27	outside	inside	standing		x					in path		Cannot fly through building though.	12			no	S	vegetation does not block	
	28	outside	inside	standing			x				in path			10			B			
	29	outside	inside	standing				x			in path			7			B			
	30	outside	inside	standing					x		in path			11				**Problem with interior of building**		F
	31	outside	inside	standing						x	in path			8				**Problem with interior of building**		F
39	32	Floor 1	Floor 2	standing									flight through floors & ceilings					N/A		NT
	33	Floor 2	Floor 1	standing														N/A		NT

Note: Two types:
PIF at Area - Planned Indirect to area. Use M79 grenade launcher from soldier to area in #33,34,&37.
Use MLRS as rocket for 35,36,&38 because flat trajectory allows testing of blockage.
AID with FO - Auto Indirect Fire with Forward Observer. Use 120 mm with regular ammo

Posture not needed for shooter.

VV	ID	Shooter Location	Target Location	Target Posture	Blocking Terrain	Blocking Vegetation	Blocking Fence	Blocking Exterior Wall	2 Interior Walls	3 Interior Walls	Visible through Window	Head Above Window	Comments	Shooter	Target	FO	Result: Acquire	Block	Result: Target Hit	Fail / NT=no test
40	1	outside	outside	standing									effect of target posture	1	18	11	Acq		K	
	2	outside	outside	crouching									No effect of posture	2	19	12	Acq		K	
	3	outside	outside	crawling										3	20	13	Acq		K	
	4	outside	outside	prone										4	21	14	Acq		S	
	5	outside	outside	foxhole										5	22	15	Acq		S	
	6	outside	outside	walking										6	23	16	Acq		K	
	7	outside	outside	running										7	24	17	Acq		K	
41	8	outside	outside	standing - partial defilade									effect of defilade	1	18	11	Acq		K	
	9	outside	outside	prone - full defilade									No effect of defilade	2	19	12	Acq		K	
	9b	outside	outside	open - standing										3	20	13	Acq		K	
42	10	outside	outside	standing	x								effect of single blockage	13	6	3	Acq	B		
	11	outside	outside	standing		x								31	26	2	Acq	no		
	12	outside	outside	standing			x							32	27	5	Acq	B		
	13	outside	outside	standing				x						33	28	4	Acq	B		
	14	outside	inside	standing					x					34	29	23	Acq	Problem inside building		F
	15	outside	inside	standing						x				35	30	24	Acq	Problem inside building		F
43	16	outside	outside	standing	x		x						effects of multiple blockages	13	6	3	Acq	B		
	17	outside	outside	standing		x	x							31	26	2	Acq	no		
	18	outside	outside	standing			x	x						32	27	5	Acq	B		
	19	outside	inside	standing			x		x					33	28	4	Acq	B		
	20	outside	inside	standing			x			x				34	29	23	Acq	Problem inside building		F
44	21	outside	inside	standing							full		target visible through window	35	30	24	Acq			
	22	outside	inside	standing							partial									
	23	outside	inside	standing							partial	yes					no acq			
	24	outside	inside	crouching							full									
	25	outside	inside	crouching							partial									
45	26	outside	inside	standing	x						in path		flight through window blocked	13	6	3	Acq	B		
	27	outside	inside	standing		x					in path			31	26	2	Acq	no		
	28	outside	inside	standing			x				in path			32	27	5	Acq	B		
	29	outside	inside	standing				x			in path			33	28	4	Acq	B		
	30	outside	inside	standing					x		in path			34	29	23	Acq	Problem inside building		F
	31	outside	inside	standing						x	in path			35	30	24	Acq	Problem inside building		F
46	32	Floor 1	Floor 2	standing									flight through floors & ceilings				N/A			NT
	33	Floor 2	Floor 1	standing													N/A			NT

Note: Two types:
PIF at Area - Planned Indirect to area. Use M79 grenade launcher from soldier to area.
AID with FO - Auto Indirect Fire with Forward Observer. Use 120 mm with regular ammo for 40,41 &44.
Use MLRS as rocket for 42, 43 & 45 because it has flat trajectory and we can test blockage.

Posture not needed for shooter.
This will be artillery against people. Later may consider other targets.

VV	ID	Shooter Location	Target Location	Target Posture	Blocking Terrain	Blocking Vegetation	Blocking Fence	Blocking Exterior Wall	2 Interior Walls	3 Interior Walls	Visible through Window	Head Above Window	Comments	Shooter	Target	Laser Designator	Block	Result: Target Hit	Fail / NT=no test	Weapon Fired?
47	1	outside	outside	standing									effect of target posture							Y
	2	outside	outside	crouching									No effect of target posture.							Y
	3	outside	outside	crawling																Y
	4	outside	outside	prone																Y
	5	outside	outside	foxhole																Y
	6	outside	outside	walking																Y
	7	outside	outside	running																Y
48	8	outside	outside	standing - partial defilade									effect of defilade	8	18	11		S		Y
	9	outside	outside	prone - full defilade									No effect of target defilade.	37	19	12		S		Y
	9b	outside	outside	open - standing										38	21	13		S		Y
49	10	outside	outside	standing	x								effect of single blockage						F	
	11	outside	outside	standing		x													F	
	12	outside	outside	standing			x												F	
	13	outside	outside	standing				x											F	
	14	outside	inside	standing					x										F	
	15	outside	inside	standing						x									F	
50	16	outside	outside	standing	x		x						effects of multiple blockages						F	
	17	outside	outside	standing		x	x												F	
	18	outside	outside	standing			x	x											F	
	19	outside	inside	standing			x		x										F	
	20	outside	inside	standing			x			x									F	
51	21	outside	inside	standing							full		target visible through window	8	18	11		S		Y
	22	outside	inside	standing							partial			9	19	12		S		Y
	23	outside	inside	standing							partial	yes		10	20	13		no acq		
	24	outside	inside	crouching							full			25	21	14		S		Y
	25	outside	inside	crouching							partial			26	22	15		S		Y
52	26	outside	inside	standing	x						in path		flight through window blocked						F	
	27	outside	inside	standing		x					in path								F	
	28	outside	inside	standing			x				in path								F	
	29	outside	inside	standing				x			in path								F	
	30	outside	inside	standing					x		in path								F	
	31	outside	inside	standing						x	in path								F	
53	32	Floor 1	Floor 2	standing									flight through floors & ceilings						F	
	33	Floor 2	Floor 1	standing															F	

DF with LD - Direct Fire with Laser Designator. Use 120 mm PGM(test)

Posture not needed for shooter.
This will be artillery against people. Later may consider other targets.

VV	ID	Posture	Terrain	Micro Terrain	Blocking Entity	Breach	Penetrate	Comments	Unit ID	Result: Average Speed	Problem	Fail / NT=no test
54	1	crawling	flat					effect of posture & terrain	18	2.0		
	2	walking	flat					Terrain Factor = 1.0	23	5.0		
	3	running	flat						22	8.0		
	4	crawling	moderate						24	2.0		
	5	walking	moderate						19	5.0		
	6	running	moderate						20	8.0		
	7	crawling	steepest						17	1.43-1.66		
	8	walking	steepest						11	3.15-3.75		
	9	running	steepest						14	2.98-4.43		
55	10	crawling	flat	road				effect of posture & terrain on road	18	2.0		
	11	walking	flat	road				Terrain Factor = 1.0	23	5.0		
	12	running	flat	road					22	8.0		
	13	crawling	moderate	road					24	2.0		
	14	walking	moderate	road					19	5.0		
	15	running	moderate	road					20	8.0		
	16	crawling	steepest	road					17	1.43-166		
	17	walking	steepest	road					11	3.14-3.75		
	18	running	steepest	road					14	2.98-5.62		
56	19	crawling	flat	grass				effect of posture & terrain on grass	18	2.0		
	20	walking	flat	grass				Terrain Factor = 1.0	23	5.0		
	21	running	flat	grass					22	8.0		
	22	crawling	moderate	grass					24	2.0		
	23	walking	moderate	grass					19	5.0		
	24	running	moderate	grass					20	8.0		
	25	crawling	steepest	grass					17	1.42-1.79		
	26	walking	steepest	grass					11	3.14-3.51		
	27	running	steepest	grass					14	2.95-5.82		
57	28	crawling	flat	woods				effect of posture on other vegetation. Terrain Factor = .5	17	2.0		
	29	walking	flat	woods				Terrain Factor = .5	11	5.0		
	30	running	flat	woods				Terrain Factor = .5	14	5.0		
	31	crawling	flat	shallow water				Terrain Factor =1.0	24	2.0		
	32	walking	flat	shallow water				Terrain Factor =1.0	19	2.0		
	33	running	flat	shallow water				Terrain Factor =1.0	20	2.0		
	34	crawling	flat	waist deep water				Terrain Factor = .75	18	.75		
	35	walking	flat	waist deep water				Terrain Factor = .75	23	.75		
	36	running	flat	waist deep water				Terrain Factor = .75	22	.75		
58	37	walking	flat	road	fence			effect of blocking, breaching & penetration	11	No B&P cap - stopped		
	38	walking	flat	road	fence	x			14	Breach	Second soldier not delayed	F
	39	walking	flat	road	fence		x		17	Penetrate		
	40	walking	flat	road	building shell				24	Stopped - cannot B or P shell		
	41	walking	flat	road	building shell	x			19	Stopped - cannot B or P shell		
	42	walking	flat	road	building shell		x		20	Stopped - cannot B or P shell		
	43	walking	flat		woods				21	vegation does not block		
	44	walking	flat	road	building wall				18	No B&P cap - stopped		

VV	ID	Posture	Terrain	Micro Terrain	Blocking Entity	Breach	Penetrate	Comments	Unit ID	Result: Average Speed	Problem	Fail / NT=no test
	45	walking	flat	road	building wall	x			22	Breach		
	46	walking	flat	road	building wall		x		23	Penetrate		
	47	walking	flat	road	exterior door				16	No B&P cap - stopped		
	48	walking	flat	road	exterior door	x			15	Breach		
	49	walking	flat	road	exterior door		x		12	Penetrate		
	50	walking	flat	road	tank				13	vehicles do not block people		
59	51	walking	flat	inside building	interior wall			movement in buildings	11	No B&P cap - stopped		
	52	walking	flat	inside building	interior wall	x			14	Breach		
	53	walking	flat	inside building	interior wall		x		17	Penetrate		
	54	walking	flat	inside building	interior door				24	No B&P cap - stopped		
	55	walking	flat	inside building	interior door	x			19	Breach		
	56	walking	flat	inside building	interior door		x		20	Penetrate		
	57	walking	flat	inside building	stairs					cannot test - no physical stairs		
60	58	crawling	external window					entering building - done with test 59	12	2.0		
	59	walking	external window						16	5.0		
	60	running	external window						15	8.0		
	61	crawling	small incline	ramp to building second story					13	.47		
	62	running	small incline	ramp to building second story					21	.95		
	63	crawling	large incline	ramp to building to 4th story					18	.35		
	64	walking	large incline	ramp to building to 4th story					22	.70		
	65	running	large incline	ramp to building to 4th story					23	.70		
61	66	crawling	flat	rubble				effect of rubble - does not vary by type of building	18	2.0		
	67	walking	flat	rubble					23	3.0		
	68	running	flat	rubble					22	3.0		
	69	crawling	moderate	rubble					24	2.0		
	70	walking	moderate	rubble					19	3.0		
	71	running	moderate	rubble					20	3.0		
	72	crawling	steep	rubble					17	1.03 -1.65		
	73	walking	steep	rubble					11	2.07		
	74	running	steep	rubble					14	3.0		

Note: if entity is Breaching or Penetration, on the screen his speed is reported as equal to his speed before he stopped at the engineering object to be breached.
If the entity cannot breach or penetrate and is therfore stopped, his speed is reported as 0.0 on the screen.

VV	ID	Entity # 1 Type	Entity #1 ID	Entity # 1 Speed	Entity # 2 Type	Entiry #2 ID	Entity # 2 Speed	Entity # 3 Type	Entity # 3 Speed	Comments	Expected Results	Result: blocked entity	Fail / NT=no test
62	1	tank		25	tank		40	tank	2km	blocking in column	#2 blocked	#2 blocked - keep playing catch and block	
	2	tank		25	tank		40	tank	3km	Modify speeds interactively	#2 & #3 blocked	#2 & #3 blocked	
	3	tank		40	tank		40	tank	2km		none blocked	none blocked	
	4	tank		40	tank		25	tank	2km		none blocked	none blocked	
	5	tank		25	tank		25	tank	3km		#3 blocked	#3 blocked	
	6	tank		25	tank		25	tank	2km		none blocked	none blocked	
63	7	tank	26	25	tank	25	40			moving toward each other	both blocked	both blocked	
	8	tank	6	0	tank	7	40			one tank stationary	tank # 1 blocks	tank # 1 blocks	
	9	tank	10	0	man	13	8			Tank blocking people	no blocking	no blocking	
	10	tank	5	25	man	14	0			People blocking tank	no blocking	no blocking	
	11	tank	9	25	man	16	8			moving toward each other	no blocking	no blocking	
	12	man	12	0	man	15	8			People blocking people	no blocking	no blocking	
	13	tank	8	0	tank	4	0			tanks very close (1 mm)	Do they move apart?	Yes	
	14	tank	2	0	tank	3	40			try different angles at which tank # 2 can move away from Tank # 1	Should be able to move away at 90 degrees.	Yes, do move away at 90 degrees or more. Results may vary depending on angle at which block occurs. Sometimes can move away at 45 degrees.	

Speed is in km.

Note: Assume that all three entities are in column formation on a road, with the order of 1,2 3, all moving in the same direction. Assume they start 10m apart.
Also have the entities processed 1, 2 and 3.

VV	ID	Test Description	Comment	Desired Results	Shooter	Results	Problem	Problem	Fail / NT=no test
64	1	Create external door that is transparent. Check LOS, LOF and kills for a bullet fired from outside seer to inside target.	Bullet proof workaround option 1	get LOS to target but not LOF or kill		For Direct Fire get LOS and therefore get LOF. For Indirect Fire bullet will be stopped. Not a good workaround.		Bullet proof glass does not work.	
65	2	Create three internal walls behind a window. Check LOS, LOF and kills for a bullet fired from outside seer to inside target.	Bullet proof workaround option 2	get LOS to target but not LOF or kill		If three walls are see-thru, for Direct Fire get LOS and therefore acquire target and automatically get LOF because had LOS. If three walls are not see-thru, do not get LOS to target and therefore for Direct Fire do not shoot. For Indirect Fire bullet will be stopped in either case. Not a good workaround.		Bullet proof glass does not work.	
66	3	Place man on ramp and test to see if can get LOS	Test LOS and LOF on ramp. **Results of 66-70 are in vv66.**	Get LOS? Can shoot him?	13	Get LOS and can shoot him.			
67	4	Place man on stairs and test to see if get LOS. Man is not physically on the stairs. A movement note "go to floor" is used to simulate going up stairs.	Test LOS and LOF on stairs	Get LOS? Can shoot him?	14	Get LOS as long as both entities are on same floor. When entity goes to next floor on activity node, no longer get LOS. Can shoot as long as on same floor.			
68	5	Create fences of several types of materials. Try to penetrate them.	Test to see if can penetrate fence and if type of material used for fence matters.	Can penetrate? Does it vary by material?	15,12,17	Can penetrate. It does not vary directly by type of material but each type of fence can be assigned a different terrain code that determines the time to penetrate the fence.			
69	6	Create stack of terrains of different sizes and with different movement rates. Place person on top of stack and move off slowly to determine how this is handled.	Test stacking of terrain	Does movement vary as man moves off one terrain onto another?	11	Speed changes as soldiers move between different surfaces.			
70	7	Does fence stop LOF? Does it depend on material of fence?	Test LOF on fences with different materials.	Does material of fence affect LOF?	16,24	Fence stops LOF as tested with planned Direct Fire at an area. It does not depend on the material of the fence.			

APPENDIX F

SUMMARY OF PROBLEMS FOUND DURING VERIFICATION and TESTING THE NEW PH ALGORITHM

Institute for Defense Analyses

A. Summary of Problems Uncovered During Algorithm Testing

In this Appendix, we present a list of problems found during algorithm testing, and report on the status of correcting each problem. We also present a list ordered by algorithm category, accompanied by an assessment of the severity of each problem.

Verification testing of JCATS was first performed using version 2.3 release of the model. A number of problems were found and corrected in Build 48 of version 3.0 release. When IDA received this version, all the vignettes were retested. Additional problems were found, some of which were corrected by LLNL in Build 51.1 of version 3.0, and others are still being worked on or evaluated. IDA verified that those problems identified by LLNL (and shown in Table 3, below) as having been fixed in Build 51.1 have been fixed.

Tables 1 and 2 summarize the problems found during setup and testing of the 70 vignettes using the version 2.3 release and Build 48 of the version 3.0 release, respectively. Each of these problems was reported to LLNL and discussed with them usually via email. The problems, numbered 1 through 34, are discussed in detail in Appendix G. This Appendix contains a description of the problems found, the scenario under which the problem occurred, LLNL's response to the stated problem, and the current status of problem resolution. Problems ranged from a question or need for clarification, to a simple error or ambiguity in JCATS documentation, to coding problems in the software.

Some statistics about the overall algorithm testing:

- Of the 70 vignettes designed, 66 were tested; 4 were found to be untestable;
- Of the total of 34 problems in the total course of testing with version 2.3 and version 3.0 releases, ten were classified as severe, six moderate, and eight minor problems in the code; 8 involved errors in the documentation; and 3 were simply questions raised by IDA during the testing;
- Under Build 51.1 of version 3.0 release, 55 of the vignettes passed after re-testing;
- Ten of these eleven remaining (and testable) vignettes failed due to three problems;
- Ten problems remain to be fixed or are currently being worked on; one is classified by IDA as severe, three moderate, and six minor.

Four of the vignettes could not be tested because it was not possible to set up a test of firing a munition between floors for certain missions. A vignette was considered to have failed if it did not produce the expected result, could not be implemented because of a problem, or revealed an associated result that was not correct. Appendix D identifies the ten vignettes that failed when re-tested under Build 51.1 of the version 3.0 release. The three problems that caused these ten vignettes to fail are:

- #25: a second system is allowed a free pass through a breach in progress.
- #30: cannot fire indirect fire mission at target line that is inside a building.
- #34: DS fire with Laser Designator does not work properly with buildings.

In the following two tables, we have separated the problems found into two groups: those found in version 2.3 and those found in Build 48 of version 3.0. The "ID" is the section number of Appendix G in which the details of the problem are discussed. In the tables, the current status of correcting the problem is indicated: if a decision to fix the problem has been made but not yet completed, the indication is "to be fixed;" if a decision has not been made to fix the problem or LLNL has not acknowledged or identified the problem, it is indicated as "on hold." All documentation problems are listed as "doc fix."

Table 1 Problems Found In JCATS version 2.3 Release

ID	PROBLEM DESCRIPTION	Fixed in B48 V3.0	On hold	Doc fix
1	The PH value reported on the datevent file does not show a change when LOS is blocked by vegetation, fence, or building.	X		
2	Cannot plan direct fire at a target.		X	
3	Cannot pre-plan a direct fire mission from a system that is in pop-up.	X		
4	The PH value reported on the datevent file does not show a change when LOS is through different sized windows. Also, a soldier who is in front of a window but whose head is above the window is shot at and killed. He should not be acquired.	X		
5	Defilade state reported may be a problem.		X	
6	The dismount pattern for 'dismount all' does not match the Simulation Manual.			X
7	A target behind a wood slat fence with PLOSB = .8 is killed more than 80% of the time. Note: PLOSB has no effect on LOF only used to say what percent of the time the target can be acquired through the fence.		X	
8	A ramp is not created as a ramp but rather as a raise highway, i.e., both ends are elevated. Also the LOS along the ramp is not as expected given the elevation.	X		
9	Reported speed of rifleman over elevated terrain is not consistent.		X	
10	Direct Support attacks with forward observer (FO) yield Indirect Fire output records in the datevent file (i.e. SA, IA, EA records) and those with laser designators yield Direct Fire output records (i.e. SD, ID, ED). The documentation implied that DS with Laser is Indirect Fire.			X
11	Direct Fire attacks produce records that do not match the descriptions in the datevent file.			X
12	The angle at which a tank can move away from another tank that has blocked it is not consistent		X	
13	Documentation of breach vs. penetration capabilities.			X
14	Documentation of Direct Support Fire with Laser Designator			X

Table 2 Problems Found In JCATS Build 48 of version 3.0 Release

ID	PROBLEM DESCRIPTION	Fixed in B51.1 V3.0	To be fixed	On hold	Doc fix
15	Documentation of how to shoot between floors				X
16	Rubble appears only on the backside of a building when it is hit on the front with artillery.		X		
17	JCATS does not check that vehicle can fit inside vehicle hole or vehicle fortification.			X	
18	Documentation for setting up counter battery missions.				X
19	LOS in the Z-direction is not available in a simulation report.			X	
20	There are inconsistencies in what blocks different types of fire missions.			X	
21	If the Elevation Report line drawn on the screen is less than ½ the terrain cell size no elevation changes are reported.		X		
22	The elevation report only gives elevation for terrain, not for ramps and buildings or fences or vegetation. It should also handle ramps and buildings.	X			
23	In the terrain editor creating a berm with plateau causes the editor to crash.	X			
24	Difficult to find the menu to allow user to enter data for a task force.			X	X
25	A second system is allowed a free pass through a breach in progress. A movement ME record should be (but is not) written out to the datevent file when entity is stopped.		X		
26	Number of problems in the terrain editor including the modify function not working properly and the loss of elevation when the nodes of a ramp are moved.	X			
27	Problems in the simulation with "clear all" function and extraneous error messages.	X			
28	For indirect fire, aim point and not the impact point is being reported on the IA record.	X			
29	The trafficability factor is not being applied in determining the speed of entities over various types of terrain.	X			
30	Cannot fire indirect fire mission using rocket at target line that is inside a building. Way model works.			X	

ID	PROBLEM DESCRIPTION	Fixed in B51.1 V3.0	To be fixed	On hold	Doc fix
31	Can block indirect fire mission using rocket at target line with a building that is one meter high. A rocket is modeled with a flat trajectory versus an arched trajectory used by an ICM.			X	
32	Cannot create maps or set altitude at post within berms in the Terrain Editor	X			
33	LOS is handled differently for vegetation vs. fences but the results are not consistent and the algorithm description seems to be incomplete.			X	
34	DS fire with Laser Designator does not work properly with buildings.			X	

The problems found are not all of the same significance. In an effort to give the reader a measure of the severity of the problems, we have categorized them in Table 3 as severe, moderate, or minor/insignificant problems in the code; a documentation problem, or a question. A “severe” problem is one that indicates that the model does not work as advertised and it is essential for operation. For example, in the version 2.3 release, the Probability of Hit (PH) was not adjusted for blockage of the target by a fence or wall. A “moderate” problem is one that is a strong candidate for a modification in JCATS. Such a problem may identify inconsistencies in how elements are handled in the model or special situations that do not result in the expected results. A ‘minor’ problem is one that does not affect the basic operation of the model. Modifications to fix minor problems are in the “nice to have” category. Errors or ambiguities in the documentation were noted by LLNL and have been, or will be, corrected in later versions of the manuals.

The following table lists the 34 problems by algorithm category and indicates the severity of the problem as S=Severe, M=Moderate, I-Insignificant, D-Documentation, and Q-Question. It also indicates those problems that are ‘on-hold’, i.e., a decision has not been made to fix the problem or LLNL has not acknowledged or identified the problem. In some cases, this is just the way the model works and one must work around it. If a problem is applicable to several algorithm categories, it is repeated under each category. Problems not associated with an algorithm are put in the ‘ General’ category, indicating that they were not associated with the testing of a specific vignette but were encountered in the general setup of tests.

Table 3 JCATS Problems by Algorithm Category

Legend: S=Severe, M=Moderate, I-Insignificant, D-Documentation and Q-Question.

CATEGORY	ID	PROBLEM DESCRIPTION	S	M	I	D	Q	Fixed by B51.1 V3.0	On hold
General	6	The dismount pattern for 'dismount all' does not match the Simulation Manual.				X		X	
General	17	JCATS does not check that vehicle can fit inside vehicle hole or vehicle fortification.			X				X
General	18	Documentation for setting up counter battery missions.				X		X	
General	21	If the Elevation Report line drawn on the screen is less than ½ the terrain cell size no elevation changes are reported.		X					
General	22	The elevation report only gives elevation for terrain, not for ramps and buildings or fences or vegetation. It should also handle ramps and buildings.		X					
General	23	In the terrain editor creating a berm with plateau causes the editor to crash.	X						
General	24	Difficult to find the menu to allow user to enter data for a task force.			X	X			X
General	26	Number of problems in the terrain editor including the modify function not working properly and the loss of elevation when the nodes of a ramp are moved.	X						
General	27	Problems in the simulation with "clear all" function and extraneous error messages.			X				
General	32	Cannot create maps or set altitude at post within berms in the Terrain Editor.	X						
LOS	1	The PH value reported on the datevent file does not show a change when LOS is blocked by vegetation, fence or building.	X						

CATEGORY	ID	PROBLEM DESCRIPTION	S	M	I	D	Q	Fixed by B51.1 V3.0	On hold
LOS	4	The PH value reported on the datevent file does not show a change when LOS is through different sized windows. Also, a soldier who is in front of a window but whose head is above the window is shot at and killed. He should not be acquired.	X						
LOS	19	LOS in the Z-direction is not available in a simulation report.			X				X
LOS	33	LOS is handled differently for vegetation vs. fences but the results are not consistent and the algorithm description seems to be incomplete.		X					X
LOF - Auto Direct Fire by Soldiers	5	Defilade state reported may be a problem.					X		X
LOF - Auto Direct Fire by Soldiers	7	A target behind a wood slat fence with PLOSB = .8 is killed more than 80% of the time.					X		X
LOF - Auto Direct Fire by Soldiers	20	There are inconsistencies in what blocks different types of fire missions.		X					X
LOF - PDF Soldier at Soldier	2	Cannot plan direct fire at a target.			X				X
LOF - PDF Soldier at Soldier	3	Cannot pre-plan a direct fire mission from a system that is in pop-up.	X						
LOF - PDF Soldier at Soldier	20	There are inconsistencies in what blocks different types of fire missions.		X					X
LOF - PDF at Area with Soldiers	3	Cannot pre-plan a direct fire mission from a system that is in pop-up.	X						
LOF - PDF at Area with Soldiers	11	Direct Fire attacks produce records that do not match the descriptions in the datevent file.				X			
LOF - PDF at Area with Soldiers	15	Documentation of how to shoot between floors				X			
LOF - Planned Indirect Fire	20	There are inconsistencies in what blocks different types of fire missions.		X					X

CATEGORY	ID	PROBLEM DESCRIPTION	S	M	I	D	Q	Fixed by B51.1 V3.0	On hold
LOF - Planned Indirect Fire	28	For indirect fire, aim point and not the impact point is being reported on the IA record.		X					
LOF - Planned Indirect Fire	30	Cannot fire indirect fire mission at target line that is inside a building. Way model works.		X					X
LOF - Planned Indirect Fire	31	Can block indirect fire mission using rocket at target line with a building that is one meter high. A rocket is modeled with a flat trajectory versus an arched trajectory used by an ICM.					X		
LOF - Auto Indirect Fire with FO	20	There are inconsistencies in what blocks different types of fire missions.		X					X
LOF - Auto Indirect Fire with FO	30	Cannot fire indirect fire mission at target line that is inside a building. Way model works.		X					X
LOF - Direct Fire with LD	10	Direct Support attacks with forward observer (FO) yield Indirect Fire output records in the datevent file (i.e. SA, IA, EA records) and those with laser designators yield Direct Fire output records (i.e. SD, ID, ED). The documentation implied that DS with Laser is Indirect Fire.				X			
LOF - Direct Fire with LD	14	Documentation of Direct Support Fire with Laser Designator				X			
LOF - Direct Fire with LD	20	There are inconsistencies in what blocks different types of fire missions.		X					X
LOF - Direct Fire with LD	34	DS fire with Laser Designator does not work properly with buildings.	X						X
Soldier Movement	8	A ramp is not created as a ramp but rather as a raise highway, i.e., both ends are elevated. Also the LOS along the ramp is not as expected given the elevation.	X						
Soldier Movement	9	Reported speed of rifleman over elevated terrain is not consistent.			X				X
Soldier Movement	13	Documentation of breach vs. penetration capabilities.				X			
Soldier Movement	16	Rubble appears only on the backside of a building when it is hit on the front with artillery.			X				

CATEGORY	ID	PROBLEM DESCRIPTION	S	M	I	D	Q	Fixed by B51.1 V3.0	On hold
Soldier Movement	25	A second system is allowed a free pass through a breach in progress. A movement ME record should be (but is not) written out to the datevent file when entity is stopped.	X						
Soldier Movement	29	The trafficability factor is not being applied in determining the speed of entities over various types of terrain.	X						
Vehicle Blocking	12	The angle at which a tank can move away from another tank that has blocked it is not consistent			X				X
Miscellaneous	8	A ramp is not created as a ramp but rather as a raise highway, i.e., both ends are elevated. Also the LOS along the ramp is not as expected given the elevation.	X						
Miscellaneous	20	There are inconsistencies in what blocks different types of fire missions.		X					X
Miscellaneous	29	The trafficability factor is not being applied in determining the speed of entities over various types of terrain.	X						

B. Testing the New PH Algorithm

In the version 3.0 release, LLNL upgraded the algorithm used for computing PH. We subsequently tested the model to ensure that the results were as expected.

The JCATS PH/PK editor creates data sets that define the effectiveness of each weapon against a specific target. For each munition-target pair, the input data include sixteen PH curves, by range between shooter and target, and by shooter and target postures. See Table 4 for sample data for these curves in the test database. The curves cover all combinations of the following shooter-target postures:

- the shooter being stationary or moving
- the target being stationary or moving
- the target in defilade or exposed
- head or flank shot.

The correct PH table is selected based on the situation. Three of these four postures are fairly straightforward. The one exception: The decision of whether to use the

defilade or exposed table is more complex and is described below. Once the correct situation is determined, the PH value for the correct range between shooter and target is obtained from the selected table by extrapolation between the range points in the table. This value is adjusted by a PH multiplier, which is also defined below.

Table 4. PH Curves for M16 Against Soldier, Extrapolated for Selected Ranges

Range (m)	SSDH	SSEH	SMDH	SMEH	MSDH	MSEH	MMDH	MMEH
0	32.00	99.00	0.00	64.00	32.00	48.00	0.00	24.00
5	31.20	97.25	0.00	62.40	31.20	46.80	0.00	23.40
10	30.40	95.50	0.00	60.80	30.40	45.60	0.00	22.80
15	29.60	93.75	0.00	59.20	29.60	44.40	0.00	22.20
20	28.80	92.00	0.00	57.60	28.80	43.20	0.00	21.60
25	28.00	90.25	0.00	56.00	28.00	42.00	0.00	21.00
30	27.20	88.50	0.00	54.40	27.20	40.80	0.00	20.40
35	26.40	86.75	0.00	52.80	26.40	39.60	0.00	19.80
40	25.60	85.00	0.00	51.20	25.60	38.40	0.00	19.20
45	24.80	83.25	0.00	49.60	24.80	37.20	0.00	18.60
50	24.00	81.50	0.00	48.00	24.00	36.00	0.00	18.00
55	23.20	79.75	0.00	46.40	23.20	34.80	0.00	17.40
60	22.40	78.00	0.00	44.80	22.40	33.60	0.00	16.80
65	21.60	76.25	0.00	43.20	21.60	32.40	0.00	16.20
70	20.80	74.50	0.00	41.60	20.80	31.20	0.00	15.60
75	20.00	72.75	0.00	40.00	20.00	30.00	0.00	15.00
80	19.20	71.00	0.00	38.40	19.20	28.80	0.00	14.40
85	18.40	69.25	0.00	36.80	18.40	27.60	0.00	13.80
90	17.60	67.50	0.00	35.20	17.60	26.40	0.00	13.20
95	16.80	65.75	0.00	33.60	16.80	25.20	0.00	12.60
100	16.00	64.00	0.00	32.00	16.00	24.00	0.00	12.00

Legend: S = stationary, M=moving, E=exposed, D= defilade, H=head shot.
First position is for shooter; second and third positions are for target.
Thus SSDH is shooter-stationary, target-stationary and in defilade, using head shot.
Note that no entries are available for flank shots; this is typical for dismounted targets.
Usually, each "Head" shot entry has a "Flank" shot counterpart.
Values are percentages and should be converted to fractions for use in the calculations.

The upgraded version of the PH algorithm is described below:

Let

H := Target Height (meters)
P := Target Partial Defilade Exposure (meters)
F := Target Full Defilade Exposure (meters)
C := Target's Current Exposure (meters)
[a...b) := The interval from `a' up to but not including `b'.
[a...b] := The interval from `a' to `b' inclusive.

For proper data: H >= P >= F
And, obviously: H >= C >= 0

Old Algorithm:

Target
Exposure PH
C PH Table Multiplier
--------- -------- ----------
[H...P) Exposed 1
[P...F) Defilade 1
[F...0] Defilade F/P

New Algorithm:

Target
Exposure PH
C PH Table Multiplier
--------- -------- ----------
[H...P) Exposed 1
[P...0] Defilade C/P

The new algorithm removes the discontinuities in the old algorithm and hits the point (Exposure=0,PH=0).

IDA tested the new algorithm in several ways. First, we placed a rifleman (the shooter) with M16 approximately 60 meters from the rifleman (the target), on flat terrain. We then placed various height solid fences in front of the target to see the effect of exposure on PH. The results of this series of tests are shown in Table 5.

Note that in our database: Soldier exposure is 1.75m for standing, .5m for partial defilade and .1m for full defilade. In our database, the PH for target standing, fully exposed, head shot is .99 at 0m and .64 at 100 m. The PH for target standing, defilade, head shot is .32 at 0m and .16 at 100 m.

Table 5. Results of Test for New Algorithm

Fence Height (meters)	Target Exposure (meters)	Probability of Hit Results (fraction)	Expected Results (fraction)	Percent Difference
1.7	.05	No LOS		
1.6	.15	No LOS		
1.5	.25	.1170	.1120	-4.46%
1.4	.35	.1650	.1568	-5.23%
1.3	.45	.2120	.2016	-5.16%
1.2	.55	.7800	.7800	0.00%
1.1	.65	.7800	.7800	0.00%
1.0	.75	.7780	.7800	0.26%
.75	1.0	.7790	.7800	0.13%
.5	1.25	.7800	.7800	0.00%
No fence	1.75	.7790	.7800	0.13%

The results for fences 1.5 meters and below were found to be consistent with the new algorithm. The question is, why is there no LOS for 1.6m and 1.7m fences? What constitutes "not seeing head?" For more discussion of this problem, see Appendix G, Problem 33.

In a second series of tests, various aspects of the target's posture were varied. The same shooter and target type were used as before. The tables below show the results for various situations affecting the calculation.

Table 6. Examples of changes in PH for Different Ranges Between Shooter and Target

Range	Target Moving	Target Height	Target Exposure	Probability of Hit Results	Expected Results	Percent Difference
20m	N	1.75	1.75	.920	.920	0.00%
30m	N	1.75	1.75	.885	.885	0.00%
40m	N	1.75	1.75	.850	.850	0.00%
50m	N	1.75	1.75	.815	.815	0.00%
60m	N	1.75	1.75	.780	.780	0.00%
80m	N	1.75	1.75	.710	.710	0.00%
100m	N	1.75	1.75	.640	.640	0.00%

Table 7. Examples of changes in PH for Shooter Still at 100m From Target

Target State	Target Moving	Target Height	Target Exposure	Probability of Hit Results	Expected Results	Percent Difference
standing	N	1.75	1.75	.640	.640	0.00%
crouching	N	.8	.8	.639	.640	0.15%
crawling	Y	.8	.8	.321	.320	-0.31%
prone	N	.25	.25	.080	.080	0.00%
foxhole	N	.25	.1	.032	. 032	0.00%
walking	Y	1.75	1.75	.319	.320	0.31%
running	Y	1.75	1.75	.320	.320	0.00%

Table 8. Examples of changes in PH for Shooter Moving at 100m From Target

Target State	Target Moving	Target Height	Target Exposure	Probability of Hit Results	Expected Results	Percent Difference
standing	N	1.75	1.75	.240	.240	0.00%
crouching	N	.8	.8	.2394	.240	0.25%
crawling	Y	.8	.8	.120	.120	0.00%
prone	N	.25	.25	.080	.080	0.00%
foxhole	N	.25	.1	.032	.032	0.00%
walking	Y	1.75	1.75	.120	.120	0.00%
running	Y	1.75	1.75	.120	.120	0.00%

Again, the results for these tests were found to be consistent with the new algorithm for computing PH.

APPENDIX G

JCATS PROBLEMS AND QUESTIONS ENCOUNTERED DURING VERIFICATION TESTING AND LLNL'S RESPONSES/RESOLUTIONS

Institute for Defense Analyses

1. Blocking of LOS

PROBLEM: In the V2.3 release, the PH value reported on the datevent file did not change when LOS was blocked by vegetation, fence or building.

TEST: Target soldier is immediately behind fence. The shooter uses a M16 and is 10m away on flat terrain. Shooter has shoot on, hold fire off, assume enemy on.

RESULTS: If the fence is 1.5 m high, the target is shot at and the PH is .955. If there is no fence, the PH is .954. (We also got .953 in one run with no fence.) If the fence is 1.6m or 1.7m, the target is not acquired. We got similar results using vegetation or a building. The slight difference in PH is probably due to our not being able to precisely locate the shooter and target. The range is reported to three decimal places but that may not be enough.

If the shooter is moved right up to the fence at .001 from the target, the PH is .982 for a 1.5m fence. If the fence is 1.6m or 1.7m, the target is not acquired even if the shooter is moved right up to the fence.

NOTES:

- Soldier exposure is 1.75m for standing, .5m for partial defilade and .1m for full defilade. In our database, the PH for target standing, fully exposed, head shot is .99 at 0m and .64 at 100 m.
- We do see a change in PH as the distance between shooter and target changes. The greater the distance the smaller the PH.
- We do see a change in PH with the change in posture or movement of the target.

LLNL RESPONSE: This has been found to be a code bug in JCATS 2.3. It will be corrected in the V3.0 release. LLNL used the opportunity of fixing this bug to upgrade the algorithm:

Let

H := Target Height (meters)
P := Target Partial Defilade Exposure (meters)
F := Target Full Defilade Exposure (meters)
C := Target's Current Exposure (meters)
[a...b) := The interval from `a' up to but not including `b'.
[a...b] := The interval from `a' to `b' inclusive.

For proper data: H >= P >= F
And, obviously: H >= C >= 0

Old Algorithm:

Target
Exposure PH
C PH Table Multiplier

--------- -------- ----------
[H...P) Exposed 1
[P...F) Defilade 1
[F...0] Defilade F/P

New Algorithm:
Target
Exposure PH
C PH Table Multiplier
--------- -------- ----------
[H...P) Exposed 1
[P...0] Defilade C/P

The new algorithm removes the discontinuities in the old algorithm and hits the point (Exposure=0,PH=0).

RETEST RESULTS: Using the V3.0 release, we placed a shooter rifleman with M16 approximately 60 meters from the rifleman target on flat terrain. We then positioned solid fences of various heights in front of the target.

Note: Soldier exposure is 1.75m for standing, .5m for partial defilade and .1m for full defilade. In our database, the PH for target standing, fully exposed, head shot is .99 at 0m and .64 at 100 m. The PH for target standing, defilade, head shot is .32 at 0m and .16 at 100 m.

The following PH values were reported under auto direct fire:

Fence Height	Target Exposure	Probability of Hit Results	Expected Results	Percent Difference
1.5	.25	.1170	.1120	-4.46%
1.4	.35	.1650	.1568	-5.23%
1.3	.45	.2120	.2016	-5.16%
1.2	.55	.7800	.7800	0.00%
1.1	.65	.7800	.7800	0.00%
1.0	.75	.7780	.7800	0.26%
.75	1.0	.7790	.7800	0.13%
.5	1.25	.7800	.7800	0.00%
No fence	1.75	.7790	.7800	0.13%

These results are consistent with the new algorithm.

STATUS: This was a problem in JCATS version 2.3. LLNL made modifications in version 3.0. IDA tested the new version and found that the problem had been corrected.

2. Planned Direct Fire at a Target

PROBLEM: In the V2.3 release, we were unable to plan direct fire missions at a target.

SOLUTION FROM IDA: In the V2.2 release, the user can plan target direct fire missions at a target without any problems. However, in the V2.3 release, the user can modify a target direct fire mission, but can't actually add one. Since it is not logical that one would do this in real life, we will just consider that it should not have been allowed in version 2.2 either. To do our test we will have to start the simulation and let the target be acquired before planning a target direct fire mission.

LLNL RESPONSE: Use behavior model.

STATUS: This was a problem in JCATS V2.3 release only because V2.2 release mistakenly allowed the user to plan direct fire missions at a target. Since it is not logical that one would do this in real life, we do not consider it to be a problem here. However, we might suggest a change in JCATS to allow a new mission like "If you see anyone in this `area' shoot them (or plan an ASAP DFAtTarget mission against them)".

3. Planned Direct Fire at a Target or Area When Shooter Is in Pop-Up

PROBLEM: In the V2.3 release, we were unable to pre-plan a direct fire mission from a system that was emplaced in pop-up mode.

LLNL RESPONSE: Both Direct Fire at Position (i.e. suppressive fire) and Direct Fire at a Target allowed the mission to be (prematurely) terminated by a pop-down request. The Direct Fire mission should take priority over the pop-down. The change will be made in the V3.0 release.

STATUS: This was a problem in JCATS version 2.3. LLNL made modifications in the V3.0 release. IDA tested the new version and found that the problem had been corrected.

4. LOS Through a Window

PROBLEM: In the V2.3 release, the PH value reported on the datevent file did not change when LOS was taken through different sized windows. Also, a soldier who is in front of a window but whose head is above the window is shot at and killed.

TEST: Target soldier is standing in a building but visible through a window. The shooter uses a M16 and is 20m away on flat terrain. Shooter has “shoot” on, “hold fire” off, “assume enemy” on.

RESULTS: We tested four sized windows: 3m high 0 offset, 1.5m high 1m offset, 1m high .5m offset, 1m high 0 offset. For the last two windows, a 1.75m soldier standing should have his head above the window. In all four cases the PH was .921 and the target

was killed. Just to make sure our walls were sufficient to block LOS and LOF we moved the target away from the window and behind the adjacent wall. In this case, the target was not acquired nor shot at.

LLNL RESPONSE: This PH problem is the same problem reported in problem 1. It is fixed by the JCATS bug fix reported earlier in response to problem 1.

The unexpected behavior of seeing an entity whose head is not visible (i.e. no LOS to the head), was caused by a bug in the logic of the LOS passing through a "portal" such as a window. The check for the head not being in view was handled incorrectly and allowed the exposure calculation to proceed with a resultant error in the calculation of the target's exposure (e.g. the target's exposure was not reduced due to the wall above the window blocking the LOS).

The fix makes the code behave as described, i.e. whole LOS is blocked when the LOS to the top of the head is blocked. It will be available in the pre-release of JCATS V3.0.

STATUS: This was a problem in JCATS version 2.3. LLNL made modifications in version 3.0. IDA tested the new version and found that the problem had been corrected.

5. Defilade State

SUB-PROBLEM #1: We were unable to put a standing soldier (1.75meters tall) in a foxhole of depth 1.5 meters, with “shoot” off and have him be in partial defilade. Instead, he always appeared in full defilade.

TEST: We placed a standing soldier in 1.5 meter foxhole and turned “shoot” off.

RESULTS: When we positioned a standing soldier in 1.5 meter foxhole and turned shoot off, he automatically went into full defilade. According the JCATS Simulation Manual p D-2, a system will go into full defilade **if possible**. However, the calculation to determine if he can go into full defilade is: If the cover provided by the terrain or prepared position is greater than or equal to the system's height minus full defilade exposure. In this case, 1.5 is less than (1.75 - .1) = 1.65. The calculation to determine if he can go into partial defilade is: if the cover provided by the terrain or prepared position is greater than or equal to the system's height minus partial defilade exposure. Thus, the system can go into partial defilade because 1.5 is greater than (1.75 -.5) = 1.25.

Is the system not put into the correct defilade state or is it not reported correctly on the state/status reports?

SUB-PROBLEM #2: We were unable to put a standing soldier (1.75 meters tall) in a foxhole of depth 1.5, with “shoot” off, and have him be in full defilade. Instead, he always appeared in partial defilade.

TEST: We put a soldier in 1.5 meter foxhole, turned "shoot" on and off, and varied his posture. [Note: different set of tests from above, with opposite set of resulting problems.]

RESULTS:

Posture	Shoot	Defilade	Remark
Standing	on	Partial	ok
Standing	off	Partial	ok because cannot go into full
Crouching	on	Partial	ok
Crouching	off	Full	ok
Prone	on	Partial	ok
Prone	off	Full	ok

According the JCATS Simulation Manual p D-2, a system will go into full defilade **if possible**. However, the calculation to determine if he can go into full defilade is: if the cover provided by the terrain or prepared position is greater than or equal to the system's height minus full defilade exposure. In this case, 1.5 is less than (1.75 - .1) = 1.65 so the standing soldier cannot go into full defilade. The calculation to determine if he can go into partial defilade is: if the cover provided by the terrain or prepared position is greater than or equal to the system's height minus partial defilade exposure. Thus, the system can go into partial defilade because 1.5 is greater than (1.75 -.5) = 1.25.

The calculations for prone and crouching soldiers show that they can go into full defilade.

SOLUTION: Simulation Manual p D-1 says when POP UP is off an entity will go into partial defilade when shoot is on and into full defilade when shoot is off, provided there is enough cover. The results here seem to bear that out. The standing soldier cannot go into full defilade so does not when shoot is off. Therefore, this is not a problem.

STATUS: LLNL was unable to recreate sub-problem #1, nor could IDA later duplicate the problem. Thus nothing more can be done about that problem except to record it, and possibly put it down to operator error. Sub-problem #2 is not a problem since the model worked as described in the manual.

6. Dismount Pattern

PROBLEM: The dismount pattern for 'dismount all' does not match the Simulation Manual.

TEST: We mounted five soldiers onto a tank and then performed a "dismount/all" at an activity node.

RESULTS: The soldiers dismounted LIFO. The first soldier went to a point 180 degrees to the rear. However, the second soldier went to the left (Counter-clockwise) of the first

and the third to the right (Clockwise) of the first. The Algorithm manual describes the dismount order as left then right but does not say relative to what. The Simulation manual Chapter 13 says the second soldier goes to the right (clockwise) the third to the left (counter-clockwise).

LLNL RESPONSE: LLNL will fix documentation in V3.0 release.

STATUS: This was a documentation problem. LLNL will fix documentation in V3.0 release.

7. Kill of Target Through Wood Slat Fence

PROBLEM: A target behind a wood slat fence with PLOSB = .8 is killed more than 80% of the time.

TEST: We placed a target soldier immediately behind a wood slat fence with PLOSB of .8. The shooter was a soldier with a M16 rifle, standing 10 meters away.

RESULTS: The target was killed the first 6 runs, not killed the next 2, and killed the next 3 runs. Should the kills work out to be 80% of the time?

STATUS: Note that the PLOSB is not used to determine LOF but rather to determine how frequently the target can be acquired. For direct fire, the shooter must acquired the target before he can shoot. Thus, for this example of PLOSB being .8, the shooter will acquire the target 80% of the time. The percent of the of the time the target is killed will depend on the PK values. Thus this is not a problem.

8. Elevation of a Ramp

PROBLEM: In the V2.3 release, a ramp is not created as a ramp but rather as a raised highway. Also, the LOS along the ramp does not seem to be as expected given the elevation.

TEST: We created a pavement section 95 meters long and then changed it to a ramp going from elevation 0 to 10 meters. The base terrain is all at 0 elevation. The ramp was not adjacent to a building, but stopped in mid-air. (The reason for this was that we wanted a long ramp so we could test elevation along the ramp.) We then tested movement of a rifleman along the ramp.

RESULTS: In the terrain editor, we could not determine the elevation at various points along the ramp. However, the ramp pop-up window confirmed the starting and ending elevations. The node at the 0 elevation node was displayed as green, as expected. In the simulation, we could not determine the elevation along the ramp. It always showed up as 0 on the right hand box. Also, if we selected the ramp, it was identified as a road of

elevation 10 meters. However, we could place a rifleman on the ramp and obtain an elevation at various places along the ramp. The elevation displayed when we moved the rifleman to the beginning of the ramp was 10 meters and continued to be this elevation all along the ramp until we were about 1 meter from the far end and there the elevation gradually went down to 0.

When we gave a rifleman a movement path along the ramp, he moved at the basically same rate he would on a road that was flat (8kph for fast, 5kph for medium, and 2kph for slow). The status reports showed his elevation as 10 meters until he reached the far end of the ramp. Our data says that the maximum grade for a rifleman is 600%. In one case we ran, the speed reported in the status report for a rifleman moving at fast speed (8.0) was 3.29 and 4.43 at the very beginning of the ramp, then 8.0 for most of the middle of the ramp with elevation at 10 meters, going down to 3.29 again at the end of the ramp with elevation of 5.72823. In one case we ran, the rifleman moving at the slow speed (2.0) would fluctuate between 2.0, 1.80 and 1.65. The speed we got in the report depending on when we elected to get the report. Sometime we got the same speed all the way across.

If the rifleman was positioned just below the ramp, he would not move. He appeared to be blocked by the ramp. In order to follow a movement path along the ramp, the rifleman had to initially be on the ramp itself. When he reached the other end of the ramp he again stopped. In this case, an error messages was displayed, saying that the change in altitude was too great (as should be expected given that the ramp stopped in mid-air). However, this is not consistent with the decline in elevation readings we were getting as we placed the rifleman closer and closer to the end of the ramp. Note that we also got lower elevation reading at this end when we displayed the status reports for the rifleman.

We also checked the LOS for the rifleman who is placed just off the beginning of the ramp. His LOS was blocked outside of the ramp, but there seemed to be LOS along the ramp for about 15 meters. Note that the rifleman is 1.75m tall. If the ramp is actually 10 meters high at this end, he should not be able to see anything. So, the LOS data we are getting is not consistent with the elevation data.

It appears from the elevation data and the movement characteristic, that we created an elevated road at 10 meters high. However, from the LOS data it seems to be a ramp.

Questions: Can we obtain the elevation along the ramp in another way? Should we expect the rifleman to slow down on this incline? Can we obtain information about LOS blockage in the Z direction? Could we be making a mistake in the way we are creating the ramp?

LLNL RESPONSE: It appears to be a bug in the software. This has been fixed in V3.0 release of the software. The terrain editor creates the ramps correctly, it is just that the simulation is not representing them correctly.

Can we obtain the elevation along the ramp in another way?
In the V3.0 release, you should be able to get the elevation for a given point in two ways. A terrain report will give you the elevation at the point clicked. And new for 3.0 is something called an elevation report, which gives you a graph of the elevation over the line specified. In your version the terrain report will report the proper elevation as the simulation is representing the ramp, however since ramps are done incorrectly the values will show you the bug. The terrain editor does not report elevation for ramps or lakes in the lower right, but just the underlying elevation posts. This is to keep up with the mouse motion, we could not continually ask what object you are on top of to find out its elevation. You will need to use the terrain editor like you have been doing.

Should we expect the rifleman to slow down on this incline?
Yes but in your version it is done incorrectly.

Can we obtain information about LOS blockage in the Z direction?
Yes, but since ramps are represented incorrectly in the simulation, it probably will not show up in your version.

Could we be making a mistake in the way we are creating the ramp?
I believe you are creating ramps correctly.

STATUS: This was a problem in JCATS V2.3 release. LLNL made modifications in the V3.0 release. IDA tested the new version and found that the problem had been corrected.

NOTE 1: In both the terrain report and the elevation report, elevation is reported on top of a building, not on the floor displayed.

NOTE 2: LLNL said that there is a problem with creating a ramp that is composed of several segments. If one segment goes off from another at an angle, there is a gap left between segments. To work around this problem, the ramp can be built by creating a terrain contour using 1 meter elevation posts and putting a road on top of the contour. The model handles road segments properly.

NOTE 3: LOS seems to go to the far side of a ramp (when looking across the ramp) rather than to the near side. LLNL says that LOS representation on the simulation screen is not created as a continuum, but is created from checks at certain intervals to see if LOS to that point is blocked or not. Thus, even though in reality the LOS should stop at the near side (since ramps are solid from the ground up) it may appear on the screen that the LOS goes further.

9. Status Report Speeds

PROBLEM: In the V2.3 release, the speed of a rifleman going through elevated terrain (contours of 1m, 2m and 10m) was checked interactively using the status report. The results were not found to be consistent. For example, occasionally, his correct, requested speed would be reported in the midst of a string of reduced speeds.

LLNL RESPONSE: We found a potential problem in the speed reported in the datevent file. It is possible that the wrong speed would be reported in the datevent if a system passed a node at the time the speed was calculated. If a system jumps over a node in its normal movement update, the distance to the node not the full distance is used to calculate the speed. This of course results in an incorrect speed.

Concerning the issue of the speed reported in the status report, no real problem was found. One answer could be different slope values. As systems move across real terrain, the slope is determined for each movement step not an average to the next node. This can result in different slopes which results in different calculated speeds.

STATUS: This was a problem in JCATS V2.3 release. LLNL made modifications in the V3.0 release. The problem in the status report will probably still be there since no change was made to address it.

10. Forward Observer and Laser Designator Direct Support Attacks

PROBLEM: Direct Support with Forward Observer (FO) missions yield Indirect Fire output records in the datevent file (i.e. SA, IA, EA records) and those with Laser Designators (LD) yield Direct Fire output records (i.e. SD, ID, ED). This way of reporting the results is not intuitive.

TEST: We designed the following DS with a FO mission: The system was a 120mm mortar firing a 120mm HE round. The FO was a rifleman and the target was a rifleman. The results gave an SA record with effect = DS, an FO record with CALL FOR DS, an IA record and an EA record with either an effect of KK or SUS. The munition was given the capability to fire Planned Indirect Fire and DS. “Shoot” was set to Off and “Hold Fire” to Off for the shooter. We varied a number of options for the systems and the munitions to see if we could ever get Direct Fire records with FO.

We also designed the following DS with a LD mission: The system was a 120 mortar firing a Copperhead round. The LD was a rifleman and the target was a M1A1 tank. The results gave an SD record with effect = DS, an OL record, an FL record, an ID record with effect = LL and an ED record with an effect of SUS. The munition was given the capability to fire only DS. “Shoot” was set to Off and “Hold Fire” to Off for the shooter.

We varied a number of options for the systems and the munitions to see if we could ever get Indirect Fire records with LD.

<u>RESULTS:</u> In both types of DS attacks, varying the capabilities of the munition seemed to have no effect on the output. For example, the 120mm HE mission was changed from Planned Indirect plus DS to Planned Direct plus DS, and the Copperhead mission from DS only to Planned Indirect plus DS. The “shoot” activity was also changed from Off to On for the shooter. In the DS with FO case, we then also fired Direct Fire missions against targets acquired by the shooter. However, for the DS with LD case, the shooter did not perform Direct Fire missions on targets it acquired.

When we varied the munition Fire Mode, we found that DS must be on to get DS attacks and Planned Indirect must be on to create artillery attacks.

Note that if one system acquires the target but a second one does not, the second one cannot have a Planned Direct mission at the target. This last situation causes a problem on the screen because there is no way to know which system acquired the target. (Answer: Use the intel report to see who has acquired which target.)

<u>QUESTIONS:</u> LLNL was asked the following questions. Answers received from discussion with LLNL are given in bold type within parentheses after the questions.

1. Will DS with FO always produce Indirect Fire output records and will DS with LD always produce Direct Fire output records? (**Yes**.) If so, what is the explanation for how one is defined as Direct and the other Indirect? (**See LLNL solution**.) Actually, the description of the datevent file groups the records as "artillery/DS/counter battery records" and "direct fire records".
2. The description of the datevent file indicates that OL and FL records for laser designator will appear for both Indirect Fire and Direct Fire. If DS with LD can produce Indirect Fire records, how is that accomplished? (**DS with LD can only produce Direct Fire records. The documentation will be corrected**.)
3. Should DS with LD also perform Direct Fire missions when shoot is on? (**No**.)
4. Why are these two DS missions treated differently? (**See LLNL solution**.)
5. Why would a DS with LD attack give effect = LL on the ID record but still produce a ED record showing suppression of the target? (**LLNL not sure why. They cannot recreate this situation but did find an error in the code that might be causing the error. Fix in next version**.)
6. We got PHPK values greater than 1.0, e.g. 4.735, on the ED SUS records for DS with LD. Should the probability of suppression be less than or equal to 1.0? (**Values are treated as 1.0 in the code even though they are reported as higher values**.)
7. How does munition Fire Mode affect what types of missions can be created? The results do not match our understanding of the definitions of the fire modes in the Vista Scenario Manual. (**This was not a problem. Another munition was being used for the missions**.)

8. How can one determine during the simulation which system acquired the target? If one tried to create a mission with a system that has not acquired the target, the model will not create it and will say it did not create it but does not say why. (**Use the Intel Report to determine who has acquired a given target**.)

CORRECTIONS TO AND QUESTIONS ABOUT DOCUMENTATION: LLNL was asked the following questions. Answers received from discussion with LLNL are given in bold type within parentheses after the questions.

Artillery/DS/Counter Battery Records:

1. The INT1 field is reported as 1 in the SA record for Planned Indirect fire and for DS with FO. What is reported here and what does the value 1 represent? (**The INT1 field reports the "rounds per trigger pull". This was left over from the JCM days.)**
2. The IA record also reports MUNAME. (**The documentation will be corrected**.)
3. With FO the T* = target data is not reported on the IA record. Is this the same as "when mission planned interactively"? (**The documentation will be corrected to indicate that with FO the target data is not reported. Also the reference to sensor guided munitions should be removed since this type record is not produce for that type mission, namely Direct Support with Laser Designator.)**
4. On the EA record EFFECT was SUS or KK for DS with FO. Only KK is listed. Can the value also be MOB and FP? (**Yes. The documentation will be corrected**.)
5. For the EA record MUNAME is reported before T*. (**The documentation will be corrected**.)

Direct Fire Records:

1. The INT1 field is reported as 3 in the SD record for Auto Direct Fire and Planned Direct fire at either a target or an area. The value is 1 for DS with LD. What is reported here and what do the values 3 and 1 represent? ? (**The INT1 field reports the "rounds per trigger pull". This was left over from the JCM days.)**
2. For the SD record, EFFECT can be DS for Direct Support. This is what we got with the DS with LD case. (**The documentation will be corrected**.)
3. The description for the ID record says that T* and the impact point are always reported, but our ID record for the DS mission with LD did not report them. It also did not report range, which should be reported unless the mission is suppressive. (**LLNL got different results here. They have made a correction in the code that may correct this problem.)** Is DS with LD suppressive? (**No**.) Also the INT2 field reported a value of 200. What does this represent? (**Here a value of 200 in the INT2 field identifies the munitions as "precision guided" or a smart munition. This was left over from the JCM days.)**

LLNL RESPONSE: Direct Support with Forward Observer is Indirect Fire because the observer passes the coordinates to the shooter and this is the same as artillery firing on an area. Direct Support with Laser Designator is Direct Fire because the LD puts a laser on

the target itself and the shooter aimed at the laser light. This is a PHPK munition firing at a target. Therefore, these two types of missions are being reported correctly. (Answers to questions are shown in parentheses in sections above.)

STATUS: The problems reported here were mainly documentation errors. These deficiencies are being corrected. As noted above, code errors were corrected in the V3.0 release of the code.

11. Planned Direct Fire Missions to an Area

PROBLEM: Planned Direct Fire attacks produce records that do not match the descriptions in the datevent file.

TEST: We created a planned direct fire mission to an area that would cause the M16 to try to fire through 3 interior walls.

RESULTS: In the following paragraph, notes from LLNL are given in bold type within parentheses.

On the ID record we got no impact data or coordinates even though the file description said it should be reported. (**This is suppressive fire and therefore the impact data will not be reported unless a target is hit. In this case, the impact location reported is the location of the target.)** Also in the Real 1,2,3 fields the coordinates of the aim point were reported and this was not expected. (**The documentation will be corrected)** Where should the impact data be reported? (**Currently impact is not reported if the target is not hit. LLNL could modify this.)** Also the PHPK value being reported is 0.0. Is this valid? (**PHPK values are not reported in this case, because it is not a PHPK munition**.)

The impact point should have occurred at the third wall since the bullet could not pass through it. The simulation picture also seemed to show the bullet going through the wall. The only way to show that the bullet did not pass through the wall was to place a target on the other side of the third wall and see that he did not get hit and then to show that if he were in front of the third wall he would be hit. ***(IDA NOTE: It turns out that the simulation shows the impact point on the screen and this can be seen if the target is far enough away from the wall.)***

QUESTIONS: LLNL was asked the following questions. Answers received from discussion with LLNL are given in bold type within parentheses after the questions.

1. The file description for ID records says that T* is for impact info and x,y,z coordinates are for impact location. When we did auto direct fire missions or planned direct fire at a target, we got target data (not impact data) in these fields. When we did planned direct fire missions at an area we got no data in these fields. Is this a problem with the description or the code? (**An ID record only shows the**

impact coordinates when the shot hits a target and the coordinates reported are the location of the target. Therefore if a planned direct fire missions at an area, i.e., suppressive fire, does not hit a target there is no T* data or coordinates.)

2. The file description for SD records does not indicate any values to be reported in the Real 1,2,3 fields. We got aim point coordinates in these fields on the planned direct fire mission to an area. The description should be modified to reflect this. We did not get these fields populated for any other direct fire missions. . (**The documentation will be corrected.)**
3. Given the problem here with aim point vs. impact point, which is being reported on the SA records (x,y,z coordinates of aim point) and IA record (x,y,z coordinates of impact or detonate point). **(SA records report aim point and IA records report impact point as stated in the documentation**.) Is it true that aim point and impact point mean different things? (**Yes**.)

STATUS: The problems reported here were with documentation. LLNL will correct the documentation.

LLNL says it would not be difficult to report impact location even when target is not hit. We might want to request this change. We might also suggest that it is confusing to have SA records report aim point location in the x,y,z coordinate fields and shooter location in fields Real 1,2,3 and to have SD records report shooter location in the x,y,z coordinate fields and aim point location in fields Real 1,2,3.

12. Blocking Movement

PROBLEM: In the V2.3 release, the angle at which a tank can move away from another tank that has blocked it is not consistent.

TEST: We gave one tank a two-node movement path from West to East going through a stopped tank, i.e., there was one node on each side of the fixed tank. Once the tank was blocked, we paused the simulation and moved the second node to various compass points around a circle center at the fixed tank in order to determine in which directions the first tank would be allowed to move. The terrain was flat, with no vegetation, roads, or engineering objects. The blocking radius for the M1A1 was set to four meters.

RESULTS: As expected, the first tank was prohibited from moving along any direction east of the North-South axis. The tank was permitted to move in any direction from West to almost due North. Problems arose, however, when we tested in the West-South quadrant: The first attempts were successful, but later on the tank would be restricted in its movements. For example, sometimes it was allowed to head nearly due South, while at other times it was restricted to more westerly movements.

QUESTION: What other factors might be considered in determining which way the tank can move? Does the direction in which the tank is facing matter? Perhaps the operator is employing movement nodes improperly?

LLNL RESPONSE: LLNL tried the blocking movement and it seemed to work. Because the blocking area is a fairly small circle, it is very sensitive to the angle of the movement path intersection. Without being able to see the blocking circle, it's hard to see how the movement path intersects the circle. A tank platoon aggregate was initially employed in order to get a straight line of tanks. The formation was set up to run east-west and then de-aggregated. The first tank's movement node was displayed on the operator's screen to allow the path to be planned directly over the center, and the movement update rate was set to one second (LLNL was not sure whether this latter measure would have any affect). LLNL re-ran IDA's experiment, and saw nearly identical results in both the West-North or West-South quadrants. LLNL attributed IDA's inconsistent results to likely operator errors involving the setting up of movement paths.

STATUS: This was a problem in JCATS V2.3. No code changes were made. The problem is likely attributable to operator error.

13. Breach Vs Penetration

QUESTION: How does JCATS decide whether to breach or penetrate a fence for which there are breach values and penetration values that use different terrain codes? When we tried the case we successfully achieved breaching, but could not find out how to cause a penetration.

Does it make sense to penetrate an interior wall? We can see that if one were to penetrate a fence he might climb over it but what does a soldier do to penetrate a wall?

LLNL RESPONSE: If a system has both breach and penetration capabilities against a terrain code, the state of the system's breach attribute determines if it breaches or penetrates. If breach is on, the system will breach. If breach is off, the system will penetrate. If a system can only breach and breach is off the system will be stopped. If the system can penetrate but not breach then it will penetrate no matter what the breach attribute.

Penetrating an interior wall may not make sense but the data can be set up so that's true. It is designed to allow a system to pass through a terrain object without creating a breach. Examples are opening a door with a key or maneuvering through wire. I guess one example (with imagination) of penetrating a wall might be a secret passageway through a wall.

STATUS: The problems reported here were with documentation. We suggested that LLNL add this information to the documentation.

14. Setting Up a Direct Support Mission With Laser Designator

QUESTION: We had several problems with setting up these types of missions: We associated a laser-designated Copperhead round with a 120mm mortar. The Copperhead is a sensor-guided munition. When we tried to set "LASER" to on for the 120mm mortar, however, we got an error message that the entity (i.e., the mortar) could not take that property. However, we were able to turn on the "Forward Observer" property with the mortar. We were able to get a helicopter with a Hellfire to work with a LD against tanks. We were also able to get a 120mm mortar with a Copperhead round to work with a LD against tanks. (We know that the 120mm mortar with Copperhead may not be legitimate.) However, we could get neither system to work against troops even though we changed the missions for system, munition and rifleman (the LD) from anti-tank to anti-troop. Finally, we could not create a planned mission with the 120mm mortar with Copperhead, because the munition type for Copperhead is "ball" and that was not option under the mortar.

Can LD be used against troops? How do the missions for system, munition and FO or LD work together? Which takes precedence?

LLNL RESPONSE: A laser designator can be used against troops but the munition must have a PHPK > 0. Hellfire and copperhead munitions are not normally fired at troops and I suspect they don't have PHPK values against troops in your database. (IDA Note: this turned out to be the case.)

It is true that a system may have an opportunity to engage a target with its own weapon, call a FO/DS mission, or laser designate for another shooter all at the same time. Given that all the attributes (shoot, FO, and Laser) are on and all the other requirements (acquisition, range, PHPK, mission etc.) are satisfied the target is selected by priority. Because system class, FO type, and designator type can all have different missions, they can each have a different priority for the same target. The one with the highest priority is selected. If there are no missions or they all have the same priority the selection is random.

STATUS: This was a problem in setting up the systems correctly. The problem was that there were no PHPK values against troops for the Hellfire and Copperhead in our database. When we added these values we were able to create direct support missions with laser designators.

15. LOF Stopped By Floors And Ceilings

QUESTIONS: We understand that LOF is stopped by ceilings and floors. How do we test this if the target inside a building gets turned vertical? Do we have to make a target area larger than the wall and if we do this what will the target area be? Where is the center of the target area; i.e. how far above the floor? (Answer is 1.25 meters according

to JCATS documentation.) If we want to test firing through interior walls, must we use planned direct fire to an area, since we cannot see through the walls? Can the soldier acquire the target by hearing the enemy in the other room? Would he then shoot toward the sound?

LLNL RESPONSE: It is true that LOF is stopped by ceilings and floors. If a planned direct mission inside a building has an area larger than the wall, some of the shoots will impact the ceiling and floor. None of the shoots will pass through the ceiling or floor. A planned direct fire mission can also be planned from the observed floor to a selected floor above or below and all the shoots will impact the ceiling or floor. Auto direct fire is only fired at acquired targets and targets can't be acquired through the ceiling or floor. Targets detected by sound are not engaged.

First the ceiling floor problem. The key for selecting the floor for planned direct fire is the number displayed in the floor select menu not the floor displayed on the screen. You must select and display the floor occupied by the shooter or the shooter can't be picked. If the shooter is on the 1st floor and you want to plan a mission to the 2nd floor first display the 1st floor by setting the floor number to 1 and then select the building. Then set the floor number to 2 but don't select the building. The mission is then planned over the displayed 1st floor but it is actually planned to the second floor. Of course the rounds are going to hit the ceiling.

STATUS: This was a problem in testing. The procedure for how to fire between floors is not documented anywhere, probably because there is no need to try to fire between floors since LOF is stopped by floors and ceilings. We tested firing between floors using the procedure described. The LOF was stopped as expected.

16. Rubble

PROBLEM: How does the simulation know which side of a building to place rubble? When we hit the front of a building with artillery (based on the impact points shown on the screen), we oftentimes seem to be creating rubble on the back of the building rather than the front. In other cases, however, we get rubble all around the building.

LLNL RESPONSE: The section of the wall that's rubbled should be the chord formed by the rubble diameter and the outside wall. That could be the whole building if it is all inside the rubble area.

You are also correct that there is a problem with rubbling the front side of a building. This is still true in the V3.0 release. We consider this to be a bug which needs to be corrected.

STATUS: Rubble is created by artillery fire. LLNL is working on the rubbling problem.

17. Bunkers And Vehicle Fortifications

QUESTIONS: What is a vehicle fortification? Does it afford protection from air attack? Must a vehicle fit inside the fortification? What is the difference between a vehicle hole and a vehicle fortification? Can the vehicle fire from inside either of these? It seems that both put the vehicle in defilade protection.

LLNL RESPONSE: A vehicle fortification has an area above the ground and a vehicle hole does not. Neither affords protection from air attack. The vehicle can fire from either. If the vehicle is in either of these structures it is considered to be in defilade. The model does not check to see if vehicle will fit inside either.

STATUS: It may be a problem that the model does not check to see if vehicle will fit inside either. This assumption may not be realistic or the burden may be on the user to make sure the vehicle will fit. Since the simulation does not give entity size information or engineering object size information, it makes it difficult for the user to make sure the every vehicle will fit. In other words, this information is all found in the Terrain file or in the Forces Characteristics file.

18. Counter Battery Missions

QUESTIONS: How does one set up a counter battery mission? Do you just add a counter fire sensor to the system and turn shoot on and hold fire on?

LLNL RESPONSE: The mission is indirect fire with a forward observer (FO). The FO has radar of the type that can detect incoming artillery. The FO capability is on for the FO entity and the shooter must have the Direct Support capability on and have a munition capable of firing an indirect fire mission.

STATUS: This was not a problem.

19. LOS in Z-direction

QUESTIONS: Is it true that the LOS fan displayed in the simulation is just for the x-y plane at a specific altitude and that it does not show the range of sight in the z-direction?

LLNL RESPONSE: It is true that the LOS fan display shows where LOS exists to a height above the terrain specified in parameter data. It may be possible to see an aircraft at an x, y coordinate above the terrain that the LOS shows no LOS exists.

LOS rays are not always precise, especially in the third dimension. LOS can pass under a bridge but not under a ramp.

STATUS: This was not an error in the code. However, it is desirable to be able to see LOS in the Z-direction. Perhaps we should propose this as a future enhancement to JCATS.

Note that during the simulation the user may set or change the height for which LOS is displayed in the simulation. Thus, he can check the LOS at various heights. The value is set separately on each workstation.

20. Testing Blockage of LOF

PROBLEM: We ran a number of tests designed to examine blocking of LOF. Several of these were initially found to be unsuccessful.

- For **auto direct fire and planned direct fire at a target** the LOS implies LOF is the rule. In other words, if LOS is obtained it is assumed that LOF is unblocked, and the flight of the bullet is not followed. Alternatively, if the blocking entities (fence, building, etc) are opaque, then LOS is blocked and LOF is blocked as well. However, if the PLOSB value for the blocking entities is 0, i.e. transparent, then there is LOS and LOF is assumed. The reasoning being that if we can see through it, we can shoot through it. We found this to be true for external walls and the third interior wall. In other words, the code works in the manner described by LLNL. Thus, this part is completed. (IDA Note: This is not a problem,)

- We tried testing the blockage of LOF during **indirect fire at an area** missions, using a grenade and a fence. The mission, however, was aborted by the model before it began. In this case, there was a 1.7m fence in front of the target area that contains a rifleman. In our database, the range of the grenade is 50 meters and the shooter and target area are 20m apart. For a range of 1 meter, the angle of fall is 30 degrees; for 25 meters, it is 40 degrees; and for 50 meters, it is 45 degrees. What data should we look at to see what the trajectory is? What does the code check against to determine that it cannot reach the target? How can we test LOF blockage in this case? (IDA Note: We never could solve this problem.)

- For **planned direct fire at an area**, the datevent file only gives the aim point, not the impact point. Therefore, even though on the screen it looks as if a building can block the fire we cannot show it through the data results. For fences it looks on the screen as if the bullet passed through the fence but it does not suppress the target located in the area at which it is aiming. For vegetation, we sometimes suppress the target in the aim area and sometimes do not. How can we test the blockage of LOF for planned direct fire at an area? (IDA Note: We eventually solved this problem by simply measuring the impact point directly from the screen. This is not a problem.)

- For **planned indirect fire**, we do not know how to set up a test for LOF blockage using the 120mm mortar. What is the trajectory? How does LOF blockage work in this case? (IDA Note: We changed the munition to MLRS used as a rocket to lower

the trajectory of the flight and more chance of the shot being blocked. We were able to get blockage this way.)

- For **planned indirect fire with an FO**, we put a three-story building between the 120mm mortar and the rifleman/target, just in front of the target. The FO rifleman was placed so that he could see the target. The FO called for DS and got it. However, the target was suppressed. Should this be the case? Does the building not block the mortar? Can we devise a test to block the mortar in this case? Also, the aim point, impact point and target coordinates seem to have the same coordinates. We have not gotten any cases where the aim point and the impact point are different. (IDA Note: We changed the munition to MLRS to lower the trajectory of the flight and more chance of the shot being blocked. We were able to get blockage this way.)

LLNL RESPONSE: Planned area direct fire works as described above. That is the way it was designed. Exterior walls always stop planned area direct fire. The original USAF requirement for tracked missed shots dealt with small arms fire inside of buildings or outside of buildings. An assumption was made (to simplify calculations) that exterior walls would stop small arms fire. Planned direct fire at an area will pass through a window in an exterior wall but not a transparent wall. So if exterior walls are made solid and windows are added to see through, auto and planned direct act the same. Planned direct fire at a target was added later and it really puts the target on the auto direct queue. If a target can be seen it can be shot at with auto direct no matter how many walls. We would need to add a wall LOF
characteristics for munitions types to solve the inconsistency you are referring to and that hasn't been done.

There is a short description of grenades in the artillery section of the 3.0 simulation manual. The part that is missing is when a munition considered to be a grenade rather than a conventional indirect munition. The criteria for grenades are:

- it is a planned indirect fire munition
- its maximum range is < 100m
- it is not a smart, sensor guided, or crew guided munition.

Because grenades can be thrown underhand or rolled as well as over hand we
don't use the conventional LOF calculations to determine if they hit terrain features. A troop can reach around a corner and throw a grenade even though he may not be able to see around the corner.

STATUS: Not being worked on.

21. Elevation Report in Version 3.0

PROBLEM: The Elevation Report provided in the V3.0 release of JCATS does not seem to work properly. When two locations that are less than 55 meters apart are selected for the report, the graph gives a flat line with the length of the x-axis being less than one meter. The scale on the graph does not cover the distance between the two locations. When the two locations are greater than 55 meters, the graph is a straight line from the elevation at the first location to the elevation at the second location. This graph does not show any change in elevation along the way between the two locations. My understanding of this report is that it should reflect the change in elevation along the path from the first location to the second location.

LLNL RESPONSE: If the Elevation Report line drawn on the screen is less than 1/2 the terrain cell size no elevation changes are reported. This is caused because the sample step distance is missing changes and small buildings when small distances are requested on large terrain files.

Having too many sample points kills performance.

STATUS: LLNL has looked at this problem. If the terrain is large, the Elevation Report will not see the building. No change is proposed by LLNL.

22. Elevation Report VS. Terrain Report in Version 3.0

PROBLEM: In Build 48 of the V3.0 release, the terrain report gave elevation for terrain, ramps and buildings but not for fences and vegetation. The elevation report only gave elevation for terrain, not for ramps and buildings or fences or vegetation. Should the elevation report also handle ramps and buildings?

TEST: We placed a building on top of a hill, and drew a line for the elevation report. We then displayed the terrain report, and clicked on points along the line for the elevation report.

RESULTS: At the building, the elevation report indicated a height of 8.1 meters, while the terrain report indicated a height of 22.8 meters. The building was 15 meters high. Thus, the elevation report did not handle buildings.

LLNL RESPONSE: In Build 51.1 of the V3.0 release, the terrain and elevation report should both handle terrain, ramps, and buildings but not fences or vegetation.

STATUS: This problem has been fixed in Build 51.1 of the V3.0 release.

23. Saving Force Files and Terrain Files in Version 3.0

PROBLEM: We often have trouble saving the force plan from the simulation. (Note: This may have just been a user error in setting up the correct pointers in the setup file.)

We have also had trouble with the terrain editor. Recently, we have had the following problems:

- We cannot access the option to turn a building shell into an enhanced building. If we have an enhanced building created in a V2.2 release terrain file, we can add windows, door, etc. to it. However, we cannot add these items to a building shell and when we select the shell, the 'Enhance buildings' option under tools was not available. Also if we selected a building shell when the building interior menu was up and the option 'select' was on, the terrain editor would stop responding and finally crash.
- When we added a berm with plateau to a terrain file and then saved the file, the feature was not saved in the terrain file. Several times I added a series of three concentric berms of decreasing size, and when I added the third berm the terrain editor crashed. Finally, it crashed and I was not able to load the terrain file into the editor. Is there a way to recover the file?

LLNL RESPONSE: The capability to turn a shell into an enhanced building is accessed by selecting "buildings" menu, selecting the shell and then selecting "add interior". This option is available in the version 3.0.

The berm with a plateau crash is a known problem.

STATUS: The berm problem has been fixed in Build 51.1 of the V3.0 release.

The problem with changing a shell into an enhanced building was that IDA was not following the new procedure. Once a shell has been enhanced, the 'Enhance buildings' option under tools was available and allowed the building to be sunk.

24. User Interface to Version 3.0 of JCATS

PROBLEM: We had to search a long time to find the menu to allow the user to enter data for a task force. Specifically, we found the only way to get to the frontage data that specifies the capability of a task force to create engineering objects was to double click on the task force name in the organizational chart. This was not intuitive and only found by trial and error. If we had not been familiar with the earlier version, we would not have known the data even existed. Perhaps a direct route to this data can be added under the organizational menu. Also this option to double click on the task force name should be well documented.

We are also having trouble locating the Global parameters for a scenario, specifically the Vehicles Block Movement flag and the Vehicles Block LOS flag.

LLNL RESPONSE: The Global parameters data can be reached either from the parameter pull down list or the Global button when "parameters" is selected.

STATUS: These notations should be made in the version 3.0 documentation.

25. Breach vs. Penetrate in Version 3.0 of JCATS

PROBLEM: During testing we observed the following: If a door is located inside a wall then the time to breach will be the minimum of the time to breach the wall or the door. Normally, the data would indicate shorter times to breach a door than a wall, but just in case the data are not logical, the minimum is used. Consequently, if the system has breach capability for the wall but not for the door, the system will still breach the door.

During testing, we observed that if a system breaches an object a yellow line will indicate where the breach is and other systems will indeed travel through the breached area and not be delayed. If a system penetrates, no indication is left in the object and other systems that travel the same route will have to penetrate the object themselves and will be delayed. This is as expected.

There is, however, a problem with breaching when the more than one system is to use the breach. If two systems have the same path through a wall that either can breach, the first to arrive at the wall will breach the wall and will be delayed at the wall until the beach is completed. If the second system arrives at the wall before the breach is completed, he will not be stopped but will be allowed the pass through the breach being created. The second system should also be delayed until the breach is completed. Both the onscreen location and the location reported in the datevent file show that the second system does indeed pass through before the breach is completed. If short breach times of 1 or 2 seconds are used it is hard to see this situation. However, if longer breach times of 30 to 60 seconds are used, it becomes obvious.

Movement records in the datevent file indicate when a system starts and ends a breaching activity or a penetration activity. However, if the system is blocked by the object a MN (movement node) record with TU (turn) is produced as the last movement record. Should not JCATS produce a ME (movement end) record with SLO (stopped by linear object)? We have also observed that the time to breach as expressed in the datevent file does not always match the time set in the input data. For example, we got 105 seconds when we were expecting 100 seconds**.**

LLNL RESPONSE: The discussion concerning the door on a wall is correct; the model is meant to work in that fashion.

The discussion regarding breaching and penetrating is also correct: Breaching leaves a permanent opening, but penetrating just opens and closes or climbs over.

The issue concerning a second system getting a free pass through a breach in progress is a problem currently be addressed. Similarly, LLNL acknowledges that a movement end (ME) record should be written when a system is stopped by the terrain, and that this is currently not be done by the model.

The difference between the breach time and the time reported in the datevent file is likely the movement update rate.

STATUS: Problems being addressed by LLNL. However, they were not yet fixed in Build 51.1 of the V3.0 release.

26. Version 3.0 Terrain Editor

PROBLEM: We have noted a number of problems in the terrain editor:

- Cannot use the modify function to change the type of exterior wall, door or window
- Sometimes the modify function will work on an interior wall an other times it will not
- If the modify function does work to change the type of an interior wall, any doors or windows in the wall are lost
- Sometimes the modify function will work on interior doors and windows and sometimes it will not
- When a building shell is converted to an enhanced building, the exterior walls are always wall type 1. How can we make them another type upon conversion rather than having to delete them and recreated them as a different type?
- It is difficult to add doors and windows to a wall. If you draw the line for a door on the line for the wall, the door or window will not be created. However, if you draw the line near the wall it will be popped into place on the closest wall.
- If you move the nodes of a ramp in the terrain editor, it loses its elevation, i.e., it reverts to a level pavement. The user has to recreate the ramp using the ramp tool.

LLNL RESPONSE: The first two items were identified as problems by LLNL, and have been fixed a later version of the code.

When a wall is modified the interior doors and windows are deleted. That is kind of a method of cleaning up problems that could get messy.

The modify function has been fixed.

When a shell building is converted, the exterior wall type can be set to other than Type 1 by selecting the wall type before adding interiors.

The trick about adding windows and doors is to have both nodes inside the building. If you try and put the nodes on the wall it's very easy to be slightly outside which is out of bounds.

Didn't plan on moving ramps.

STATUS: Problems fixed in Build 51.1 of the V3.0 release.

27. Version 3.0 Simulation Interface

PROBLEM: We have noted a number of problems with the simulation interface:

1. When clear all or close all does not work with any of the reports. It does work with the pull-down menus
2. The error message "unable to create engineering object" appears frequently and we have not isolated under what circumstances.
3. LOS on the simulation screen does not consider the PLOSB values of the engineering objects, i.e., the LOS displayed only considers if an object is in the way, not whether one can see through it.
4. Sometimes "clear all" leaves parts of symbols on the screen, e.g., movement nodes.

LLNL RESPONSE: LLNL is aware that "close all reports" doesn't close all report windows. LLNL plans to remove the Clear button from the report menu.

True the "unable to create engineering objects" appears occasionally. This seems to happen when using a controller client. Several menu functions (engineering, movement routes, etc.) require that the controller only have one Force displayed.

The Planning LOS function does consider if an object can be seen through. LOS through windows is sometimes seen if the geometry and sample spacing happen to be correct. LOS through a fence will show the oblique angle limits.

Garbage left behind after a Clear All are eventually cleared when a zoom is done.

STATUS: Problems being addressed by LLNL. Items 1,2 and 4 have been fixed in Build 51.1 of the V3.0 release. Item 3 is true but is not considered by LLNL to be a problem.

28. Indirect Fire

PROBLEM: It appears that the aim point rather than the impact point is being reported on the IA record of the datevent file. When we used planned indirect fire, the hits appear on the screen all around the target area. The distance from the target area depends on the aiming errors and ballistic errors for deflection and range. If these are large the spread of hits is wide, if small the spread is small. However, for each hit the "impact point" reported on the IA record in the datevent file always matches the "aim point" reported on the SA record. The aim points vary slightly over attacks but do not have the wide range shown on the screen for the hits. Therefore, we assumed that the hits on the screen were impact points. The reported impact points, however, did not seem to correspond with the impact points shown on the screen**.** We believe the aim point is being reported on the IA record.

LLNL RESPONSE: True the IA record is showing the aim point not the impact point. Problem has been fixed in LLNL Version 3.0.

STATUS: Problem has been fixed in Build 51.1 of the V3.0 release.

29. Trafficability Factors in Movement in Version 3.0

PROBLEM: In the V2.3 release, we were able to see a reduction in the speed of a soldier traveling over terrain that had a trafficability factor less than 1.0. In the V3.0 release, we did not get a reduced speed as reported in the simulation status report. For example, we had a trafficability factor of .5 for dismounted systems on vegetation called woods. For a soldier, the basic fast speed when standing was 8 kph and the maximum is 10 kph. Thus the soldier should have slowed to 5 kph when traversing woods(i.e., .5 * 10).

LLNL RESPONSE: In Build 48 of the V3.0 release, the trafficability attributes values were being ignored. This has been fixed.

STATUS: Problem has been fixed in Build 51.1 of the V3.0 release.

30. Indirect Fire at Buildings

PROBLEM: We cannot fire indirect fire mission at a target line that is inside a building.

QUESTION: If planning an indirect fire mission and the target line is the center of a building, is the mission automatically aborted or is the target line given the z-coordinate of the building height?

TEST: We used 4 MLRS ICM as rockets firing at a building whose front edge was 1, 1.13, 1.33, 1.5 km, respectively, from the shooters. The range of the MLRS was 1.0-3.2 km. We could fire all four rockets at the front edge of the building, at the back edge of the building or beyond the building. However, when we made the target line just inside the front wall or in the middle of the building, we would get "mission aborted, target out of range" error messages for all 4 shooters. We would get the same problem other places inside the building.

We tried leaving the target line at the middle of the building (keeping the same force file with the missions specified when the building was there), but removing the building and creating a new terrain file. In this case, we would again get the "mission aborted, target out of range" error messages for all 4 shooters. However, if we deleted the missions and recreate them at the same target line (when there is no building there), all four shooters would fire and no error message would be displayed.

We also noticed that, in the case just described, the impact point always appeared on the operator’s screen at the front edge of the building, regardless of whether we were aiming at the front edge or the back edge of the building. The building was 3 meters high and 100 meters deep from front edge to back edge. The MLRS rounds had a 10-degree angle of fall. Assuming that the front of the building was blocking the trajectory, then what was observed was in fact correct: We found rubble both in front and in back of the building.

LLNL RESPONSE: The problem with not being able to plan a mission on a building occurs when the target appears to be inside the building. The model will not let you plan a mission from outside a building to the inside. It will allow you to plan a mission inside if the shooter is also inside. To plan a mission onto the roof of a building, the current floor number selected in the Building, Floor menu must be **greater than** the roof number. It doesn't matter which floor is displayed. For example, a building with 4 floors has the roof on the 5. To fire a mission on the roof, the floor number must be set to => 6. There is a special case of throwing a grenade against a building. It is possible that the calculated path could put a grenade through a window but unless the window is very large it is unlikely.

STATUS: We found no problem with he way the model works in this case. However, the question remains about why the missions have to be recreated when the terrain file is changed to remove the buildings.

31. Blocking of Indirect Fire by Buildings

PROBLEM: We can block an indirect fire mission at a target line with a building that is one meter high. We expected that the building would have to be more than 50 meters high to block the munition in this test.

Test: We were testing the blocking of LOF of MLRS ICM, used as a rocket with a 10-degree angle of fall. The shooter was placed 1.45 km from target area. Buildings 50 meters, 15 meters, and 1 meter high were located .07 km, .185 km and .3 km from the desired impact point, respectively.

When we ran the simulation, the impact point was on the 1-meter building and we got rubble around this site. This building was within the range of 25% of the desired impact point (1.45 * .25 = .36). However, a plot of the angle of fall indicates that the round should only hit a building that is taller than 50 meters. When we removed the 1-meter building, the round hit the 15-meter building.

We drew the angle of fall line back from the desired impact point and looked to see what size building would intersect the line within either 1000m or 25% of the range (whichever value is smaller). Our picture looks something like this:

```
 x
 x
 x angle of fall
IxxxxxxxBxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxS
 bldg
Impact Shooter
```

Is this being set up incorrectly? Or is the code not checking the height of the building, but instead only checking that the building is within the last 25% of the range?

LLNL RESPONSE: JCATS backups the minimum of (1000 meters or 25% of the range) from the impact point along the angle of fall. The first building intersected within that distance should be the one hit. LLNL tested a 155SP shooting an HE round over a 1-meter building from approximately 8 km away. The round had approximately a 11 degree AOF at that range and all the CEP errors were set to 0. The building was about 30m wide. When the aim point was placed at varying distances beyond the target the round never hit the near side of the building. When the aim point was placed as close as possible to the far wall, the round landed on the edge of the roof nearest the aim point. (IDA Note: LLNL used ballistic guidance.)

STATUS: This problem was one of communication. IDA used the MLRS as a rocket and LLNL used it with ballistic guidance. The rocket is modeled in JCATS as having a flat

trajectory and thus it will be blocked by any size building. The trajectory used for ballistic guidance is an arc and JCATS does look at the last 1000 meters or 25% of the range. Thus, this was not a problem.

32. Creating Maps with the Terrain Editor

PROBLEM: We could not create a map in Build 48 of the V3.0 release. We kept getting message that New Map must fit into global map.

The Global map is 2 km on each side with 20 resolution (i.e. post every 100 meters). We started a map at the same left corner and used 1km and 10 resolution. This should put the new map on top of a portion of the global map with the same posts. We tried this only after being unable to get the terrain editor to create a .1-km map with 10 resolution (posts every 10 meters).

Also, we could not create a berm or plateau in which the operator set the altitude, nor could we set selected posts inside the global map.

LLNL RESPONSE: Problem recognized and corrected in later version of the code.

STATUS: Problems fixed in Build 51.1 of the V3.0 release.

33. Vegetation vs. Fences in LOS Algorithm

PROBLEM: LOS is handled differently for vegetation vs. fences; but, the results are not consistent, and the algorithm description seems to be incomplete.

QUESTION 1: Shooter and target are 60 meters apart and the target has 1.5 meters of opaque scrubs in front of him (PLOSB=1). With auto direct fire using an M16 rifle, the shooter hits target with PH of .78. This is the same PH for our data as if there were no scrubs in front of the target. If a 1.5 m fence is put in front of the target the PH is .1120, as expected based on the new blockage algorithm.

If the scrub is raised to 1.7m, the PH is still .78, but at 2m the shooter does not acquire the target.

If the fence is raised to 1.6 or 1.7m, the shooter does not acquire the target.

Should the same results be obtained no matter whether the target is blocked by dirt, vegetation, fences or buildings?

LLNL RESPONSE: The difference in the PH is because of the amount of the target exposed. When a target is seen through only vegetation, the entire target is potentially exposed so the size is based on the angle from the ground to the top of the target. With a fence the angle is from the top of the fence to the top of the target.

QUESTION 2: According to the LOS algorithm, there are two issues: attenuation and exposure. Vegetation affects attenuation and terrain, buildings, etc. affect exposure. The procedure is:

1) Cast a ray from sensor to the head of the entity to be acquired.
2) Cast a ray from sensor to the foot of the entity to be acquired.
3) For each ray, make a list of polygon intersections
4) If the head ray is blocked at any intersection, then the entity cannot be seen.
5) For each intersection, get attenuation and new effective foot ray.
 - attenuation for linear objects is a function of viewing angle
 - if old foot ray is blocked by terrain feature, try elevating it to clear obstacle. If exposure is still greater than 0, this is the new effective foot ray. Else, the entity cannot be seen.

This accounts for the difference in PH for vegetation vs. fence, since vegetation does not affect exposure. When the vegetation gets to 1.75m high the target is blocked (in my scenario) because the head ray is blocked. But doesn't the model have to check here to see what PLOSB is too? In our scenario, if the PLOSB = 1 for the vegetation, then we get no LOS; but, if PLOSB = 0, we do get LOS even though the vegetation is the same height as the target (1.75 meters).

In other words, if any part of a target can be seen, the PLOSB is not considered. On the other hand, if the target cannot be seen (i.e., head ray intersects vegetation obstacle), then the PLOSB is considered.

However, the algorithm does not explain why we could not acquire a target behind either a 1.6-meter or a 1.7-meter fence, but we could acquire one behind a 1.5-meter fence.

LLNL RESPONSE: LLNL tested troops 1.75 meters in height, looking over 1.5, 1.6, 1.7, and 2 meter fences. They acquired each other over all but the 2-meter fence.

STATUS: Problems was addressed by LLNL. They do not see a problem here. However, IDA still cannot acquire the target over a 1.6 or 1.7 meter fence.

34. Direct Support Fire with Laser Designator Mission Interaction with Buildings

PROBLEM: Direct Support (DS) fire with Laser Designator (LD) does not work properly with buildings. The shooter is able to attack a target directly behind a building and it can also attack a target inside a building.

TEST: During testing of DS fire with Laser Designator, we came across two unusual situations:

(1) In the first case, we placed the LD and the target on the backside of a 99-meter high building, right next to the wall on the ground, so that both were blocked from the DS shooter by the building. The shooter was firing a Copperhead round about 1000m from

the aimpoint. The LD lased the target, the shooter fired, and the target was suppressed. In other words, the building failed to block the trajectory of the munition. How is it possible for the round to travel along such a trajectory that it both clears the building and comes straight down behind the building? Also, how does the shooter pick up the laser which is only visible for 120 degrees?

(2) In the second case, we placed the LD and the target inside a 15m building, on the first floor. Again, the LD lased the target, the shooter fired the Copperhead round, and the impact point appeared on the screen at the location of the target. How can the shooter pick up the laser inside a building?

LLNL RESPONSE: There is a problem with Direct Support fire, laser guided munitions not being blocked by the terrain. There is a LOF check that is failing. Because DS fire, laser guided munitions have no ballistic or angle of fall data they are a special case. The direct fire, laser guided munitions (those where the entity lasing/guiding and firing the round are one and the same) move along a LOS path and are pretty straight forward and they do work. LLNL considers this to be a bug in the code, although it has not been fixed in Build 51.1 of the V3.0 release.

For the case of LD into a building, if the entity with the laser can see the target through a window or breach, it can designate. Upon the designating entity's request, the shooting entity fires the copperhead knowing nothing about the target. The Copperhead has no angle of fall information because it is a guided munition not an indirect fire HE or ICM. The path is approximated using 45 degrees. If the target is inside a building, LOS is lost and a ID LL record is written to the datevent file. Unfortunately we don't have another impact point so we use the original aim point. The round lands at the aim point and the **suppression** effects are accessed. The target is not evaluated for the PHPK.

STATUS: Adding a check on LOF for the DS with LD should be done but it is hard. Problems being addressed by LLNL. The problems are:

- LOF is not checked for Direct Support with Laser Designator when a building is in the path
- When the LOF for Direct Support with Laser Designator is into a building the angle of fall is assumed to be 45 degrees.

APPENDIX H

SUGGESTIONS FOR HANDLING DIFFICULTIES

Institute for Defense Analyses

In the course of testing the vignettes, we recorded all difficulties we encountered, with the thought that users could benefit from our suggestions for handling these difficulties. They are discussed in the following paragraphs by topic.

A. Creating Ramps

In the Terrain Editor, a ramp must be created starting from a linear pavement segment, not a polygon. To see the starting end of a ramp, go into node mode. The starting end will be green. In order for a system to enter a building from a ramp, the ramp must touch the building and the maximum vertical step allowed for the system must exceed the incline of the ramp.

If the user moves the nodes of a ramp in the Terrain Editor, the ramp loses its elevation, i.e., it reverts back to level pavement. The user then has to recreate the ramp using the ramp tool. However, if he moves the entire ramp using the translate function the ramp retains its elevation.

LLNL said that there is a problem with creating a ramp that comprises several segments. If one segment goes off from another at an angle, it leaves a gap. To work around this problem, the ramp can be built by creating a terrain contour using one meter elevation posts and putting a road on top of the contour. The model handles road segments properly.

B. LOS and Ramps

LOS seems to go to the far side of a ramp (when looking across the ramp) rather than to the near side. LLNL says that LOS representation on the simulation screen is not created as a continuum, but is created from checks at certain intervals to see if LOS to that point is blocked or not. Thus, even though in reality the LOS should stop at the near side (since ramps are solid from the ground up), it may appear on the screen that it goes further.

C. Knowing Who Has Acquired Target

For Planned Direct Fire at the Target missions, the only systems that can be used to fire on the target are those that have actually acquired it. Use the Intel report to see who has acquired the target and then plan the mission accordingly.

D. Planned Indirect Fire Missions Against Buildings

If the user plans an indirect fire mission to a target line that is on top of a building, when the model tries to simulate the mission the user will receive the error message

"mission aborted, target out of range." See Appendix G, Problem #30, for a detailed discussion of this problem.

According to LLNL, this problem occurs because the target appears to the model to be inside the building. The model will not let the user plan a mission from outside to inside a building. To overcome this problem and plan a mission onto the roof of a building, the current floor number selected in the Building, Floor menu must be greater than roof number of the building. Beyond that restriction, it does not matter which floor number is displayed. For example, a building with four floors has the roof on the fifth. To fire a mission on the roof, the floor number must be set to a value equal to or greater than six.

E. Firing Between Floors

A special setup is required to fire between floors. The key for selecting the floor for planned direct fire is the number displayed in the floor select menu, not the floor displayed on the screen. Specifically, the user must select and display the floor occupied by the shooter or the shooter cannot be picked. If the shooter is on the 1st floor and the user wants to plan a mission to the 2nd floor, he must first display the 1st floor by setting the floor number to 1 and then select the building. Then he sets the floor number to 2, but does not select the building. The mission is then planned over the displayed 1st floor, but it is actually planned on the second floor.

F. Breach and Penetrate

The following table summarizes how JCATS handles breach vs. penetrate. The breach and penetration capabilities are set in the parameters data for the breaching system type versus the engineering object. Breach on or off is set in the simulation for the specific system.

Set Breach	Breach Capability	Penetration Capability	Results
ON	Yes	Yes	Breach
ON	Yes	No	Breach
ON	No	Yes	Penetrate
ON	No	No	Blocked
OFF	Yes	Yes	Penetrate
OFF	No	Yes	Penetrate
OFF	Yes	No	Blocked
OFF	No	No	Blocked

Thus, if Breach is set to on, and if the system has the capability to breach the engineering object (e.g., wall, fence, door, window), then it will breach. Otherwise, if it has penetration capability, the system will penetrate the object. If it has neither breach nor penetration capability against that specific object, it will be blocked. The time to breach or penetrate is associated with the breach code or penetration code.

If a door is located inside a wall, then the time to breach will be the minimum of the time to breach the wall or the door. Normally, the data would indicate shorter times to breach a door than a wall, but just in case the data are not logical, the minimum is used. Consequently, if the system has breach capability for the wall but not for the door, the system will still breach the door.

During testing, we observed that if a system breaches an object, a yellow line will indicate where the breach is and other systems will indeed travel through the breached area and not be delayed. If a system penetrates, no indication is left in the object, and other systems that travel the same route will have to penetrate the object themselves and will be delayed.

There is, however, a problem with breaching when more than one system is to use the breach. A second system gets a free pass through the breach while the breach is in progress. See Appendix G for a detailed description of this problem.

Building shells cannot be breached or penetrated. If a system's path goes through the building shell, the system will be stopped at the shell wall and an error message will be displayed: "unit blocked by elevation step difference." This message is given because systems may not enter building shells, but they may go to the roof of the shell. JCATS is trying to move the system to the roof but cannot, because the system cannot make that big of a vertical step. However, the stoppage is expected, as shells have no breach codes or penetration codes. If the user wishes to allow a system to breach or penetrate a shell, he can convert the shell to an enhanced building by adding interior. To control the wall type used, the user should select the shell, then select the wall type, and then select "add interior." The user may add other doors, windows, and interior walls. The breach and penetration codes associated with the type wall for the specific system type will govern whether the system can breach/penetrate and how long it will take.

G. Terrain Editor Hints

When a building shell is converted to an enhanced building, the exterior walls will default to wall type 1. To specify another type of exterior wall, the user first must select the shell to convert, then select the wall type, and finally select the "add interior" option. The user may then add windows , doors, and interior walls.

It is difficult to add doors and windows to a wall. If the user draws the line for a door on the line for the wall, the door or window will not be created. This occurs because the line for the door cannot cross the line for the wall; it is difficult to prevent this. However, if he draws the line near the wall, it will be popped into place on the closest wall.

H. Fitting Vehicles in Vehicle Holes and Fortifications

The model does not automatically check to see if a vehicle will fit inside either a vehicle hole or vehicle fortification. The burden is on the user to make sure the vehicle will fit. Since the simulation interface does not give entity size information or engineering object size information, it makes it difficult for the user to make sure every vehicle will fit. This information is found in the Terrain file or in the Forces Characteristics file. Although difficult, the user should check the sizes for the logical consistency.

I. Setting Up Various Types of Fire Missions

Direct Support with Forward Observer is considered to be a type of Indirect Fire mission because the observer passes the coordinates to the shooter, similar to artillery firing on an area. On the other hand, Direct Support with Laser Designator (LD) is considered to be a type of Direct Fire mission because the LD puts a laser on the target itself and the shooter aims at the laser light. Therefore, the round's effect is calculated using PHPK data. The user should be aware that when setting up Direct Support Fire with a Laser Designator, the munition to be used must have in the JCATS database PHPK values against the target type.

Appendix C contains a table indicating the data required to set up various types of fire missions.

J. LOS vs. LOF

The user should be aware that there is an inconsistency in the way fire missions treat transparent walls. Normally it is best to have walls with PLOSB set to 100 percent so that the fire missions are consistent.

For auto direct fire and planned direct fire at a target, "LOS implies LOF" is the rule. Therefore, if the shooter gets LOS, it is assumed that he gets LOF and the flight of the bullet is not subsequently followed. If the blocking entities (fence, building, etc) are solid, then LOS is blocked as well as LOF. However, if the PLOSB value for the blocking entities is 0, i.e., the object is transparent, then there is LOS, and LOF is assumed: the shooter fires through the object. The reasoning behind this rule being that if

the shooter can see through it, he can shoot through it. An identical situation pertains in the case of transparent external walls and the third interior wall.

On the other hand, for planned direct fire at an area, external walls and third interior walls will both block LOF, even if the walls are transparent. This is the way JCATS was designed. Exterior walls always stop planned area direct fire. The original USAF requirement for tracked missed shots dealt with small arms fire inside buildings or outside buildings. An assumption was made (to simplify calculations) that exterior walls would stop small arms fire. Planned direct fire at an area will pass through a window in an exterior wall, but not a transparent wall. So if exterior walls are made solid and windows are added to see through, auto and planned direct act the same. Planned direct fire at a target was added later, and it really puts the target on the auto direct queue. If a target can be seen it can be shot at with auto direct no matter how many walls there are. LLNL would need to add wall LOF characteristics for munitions types to solve this inconsistency.

K. Putting Buildings Close Together

In the Terrain Editor, it is sometimes difficult to place buildings close together. The easiest way to accomplish this is to go into "node" mode, zoom in, and move the nodes of the buildings to the desired distance apart.

L. Duplicating Systems

When using the option "Duplicate System" in the Vista Editor, be sure to check the "movement on slope" data in the duplicate system. The data may be different from the original system, or they may not have been copied over and therefore are set to zero.

M. Using the Elevation Report on Large Terrains

If the Elevation Report line drawn on the screen is less than one-half the terrain cell size, no elevation changes are reported. This is because the sample step distance is missing changes and small buildings when small distances are requested on large terrain files. If the terrain is large, the Elevation Report will not see the building. The user should be aware of this.

N. Using Grenades

Because grenades can be rolled or thrown underhand as well as thrown overhand, JCATS does not use the conventional LOF calculations with grenades to determine if they if they are blocked by terrain features. A soldier can reach around a corner and throw a grenade even though he may not be able to see around the corner.

There is a short description of grenades in the artillery section of the version 3.0 release simulation manual. One element missing from this discussion is how JCATS determines whether a munition is a grenade or simply a conventional indirect fire munition. The criteria for grenades are:

- it is a planned indirect fire munition;
- its maximum range is < 100m;
- it is not a smart, sensor-guided, or crew-guided munition.

O. Using Direct Support Fire with Laser Designator Against Targets Near or in Buildings

The user should be aware that when employing Direct Support Fire with Laser Designator missions, he may not get PHPK results, but only suppression of the target. See Problem #34 in Appendix G for a fuller discussion of this issue. A brief description of what may happen follows.

The indirect laser into a building is complicated. If the entity with the laser can see the target through a window or breach, it can designate. Upon the designating entity's request, the shooting entity fires a laser-guided round, e.g., Copperhead, knowing nothing about the target. The Copperhead has no angle of fall information because it is a guided munition, not an indirect fire HE or ICM round. The path is approximated using 45 degrees. If the target is inside a building, LOS is lost and a ID LL record is written to the datevent file. Unfortunately, the model does not have another impact point so it uses the original aim point to assess the rounds effects. The round is assumed by the model to have landed at the aim point, and the suppression effects alone are determined. The target is not evaluated to determined whether it was killed or wounded.

P. Use of JCATS Post-Processor

IDA has developed a JCATS post-processor to aid in the analysis of JCATS runs. The post-processor is a Microsoft Access application that is menu driven. It processes the datevent file produced by JCATS and provides numerous reports.

The JCATS post-processor allows the user to load in any number of datevent files from different JCATS runs and compare the results. The runs files may be from different runs of the same scenario or from different scenarios. When the user elects to load a particular events file, the application loads the data from the datevent file into a working table, EVENTS. Then a scenario ID and a run number are added to the records, based on user input. Then, for each record type of interest, the data from that type record are stored in a separate table, e.g., DS records are stored in table Direct_Shots. The datevent file

records have a lot of redundant data about the shooter and the target. To reduce this, we use the shooter unit ID and the target ID in the record tables to identify the shooter and target. All of the other data are stored in the tables LU_Shooter and LU_Target. A number of queries and forms have been built to display the results by run, by scenario or across several scenarios. For example, we compute Number of Kills by Type, Number of Losses, Last State of Entities, and the Range of PH values.

The post-processor was designed originally for this project. It will be improved and expanded, depending on the needs of IDA personnel. The post-processor is not currently available for general use outside IDA.

APPENDIX I

PROPOSED CHANGES TO JCATS

Institute for Defense Analyses

A. Previously Proposed Changes to JCATS

Appendix J contains a list of JCATS changes proposed by IDA as of May 7, 2001. The list was developed, in part, on MOUT ACTD requirements. A number of these proposed changes cannot be implemented, as indicated in the notes. Most items are still pending and have not moved to a high priority. The list is included as a history of proposed changes.

However, some items are highly desirable. At the September 16-17, 2000 meeting at LLNL, we discussed the then "current" list of enhancements and created a prioritized list of enhancements for MOUT. The top priorities are:

1. Robot mobility class
   - tethered to a person, concept of ownership
   - sensors off when energy runs low
   - ability to represent different sized robots with very different trafficabilities
   - represent one robot throwing another?
2. Trigger nodes to assist planning process (e.g., turn "shoot on" at this point of movement path)
3. Path creation tool
   - creates a track that any entity can follow – just need to position the system on the start of the path, and it will follow. Will be good for patrol routes.
4. Ability to ascend the exterior of a building, ladders, fire-escapes
5. Ability to create a hole in a wall using a munition, rather than breaching
6. Representation of a through-wall sensor
7. Laser designator that can be associated with area directed fire.

As of July 2001, items 1, 2, 3, and 6 had been implemented in the version 3.0 release of JCATS, while items 4, 5, and 7 were still being addressed.

B. Additional Proposed Changes to JCATS

In the course of testing the vignettes, we identified a number of changes that might benefit JCATS users; these are discussed in the following paragraphs.

1. Pre-planned ASAP Direct Fire at a Target

Currently, "planned direct fire at a target" missions cannot be planned until the shooter has acquired the target. An analyst, however, might want to pre-plan such a mission generically before starting the game, according, perhaps, to the following criteria: "If you see anyone in this area, fire at them; i.e., plan an ASAP Direct Fire mission against them."

2. Report Impact Point for Planned Direct Fire at Area

Currently, an ID record only shows the impact coordinates when the shot hits a target. Moreover, the coordinates reported are those for the location of the target. Alternatively, if a planned direct fire at an area (i.e., suppressive fire) mission does not hit a target, no impact data or coordinates are reported. It is desirable for analysis purposes to know the impact point of all direct fire missions. If reported, it can be used, for example, to document when a shot is blocked.

3. Modification to Datevent File

We believe that it is confusing to have SA records report aim point location in the x,y,z coordinate fields and shooter location in fields Real 1,2,3, and to have SD records report shooter location in the x,y,z coordinate fields and aim point location in fields Real 1,2,3. We suggest that these records formats be modified to be more consistent.

4. Obtain Size Of Engineering Objects During Simulation

There is no way to obtain the size of engineering objects (e.g., the size of a foxhole) from the simulation interface. This would be a nice feature to add. We also cannot get information on other terrain features during the simulation, for example, PLOS of an object or height of an object such as a fence.

Likewise, a difficulty obtains in regards to vehicles and their related engineering objects. A vehicle fortification has an area above the ground and a vehicle hole does not. Neither affords protection from air attack. The vehicle can fire from either location. If the vehicle is in either of these structures, it is considered to be in defilade. The model does not automatically check to see if vehicle will fit inside either structure. Since the simulation does not give entity size information or engineering object size information, it makes it difficult for the user to make sure every vehicle will fit. In other words, this information is all found in the Terrain file or in the Forces Characteristics file. We suggest that either the model check that the vehicles fit or that the sizes be available to the user from the simulation.

5. Obtain Report of LOS in the Z-direction

The LOS fan display shows where LOS exists to a user-specified height above the terrain. This height is specified in the parameter data. LOS rays are not always precise, especially in the third dimension. For example, LOS can pass under a bridge but not under a ramp. It also may be possible to see an aircraft at an x,y coordinate above the terrain that the LOS shows no LOS exists. It is desirable to be able to see LOS in the Z-direction in the simulation interface.

6. Add Pull-down Menu to Access Data for Task Force

In the Vista Editor, we had to search a long time to find the menu to allow the user to enter data for a task force. Specifically, we found the only way to get to the frontage data that specifies the capability of a task force to create engineering objects was to double click on the task force name in the organizational chart. This was not intuitive and found only by trial and error. If we had not been familiar with the earlier version, we would not have known the data even existed. Perhaps a direct route to these data can be added under the organizational menu. Also, the option to double click on the task force name should be well documented.

7. Solve Inconsistency of Auto Direct Fire and Planned Direct Fire at an Area when Passing Through Transparent Walls

In the development of JCATS, an assumption was made (to simplify calculations) that exterior walls would stop small arms fire. Planned direct fire at an area will pass through a window in an exterior wall, but not through a transparent wall. So if exterior walls are made solid and windows are added to see through, auto and planned direct fire act in identical fashion. Planned direct fire at a target was added later and it really puts the target on the auto direct queue. If a target can be seen, it can be shot at with auto direct no matter how many walls. Thus, auto direct fire will pass through transparent walls. We suggest that wall LOF characteristics for munitions types be added to solve the inconsistency of travel through transparent walls, both exterior and interior.

APPENDIX J

PREVIOUSLY PROPOSED CHANGES to JCATS

Institute for Defense Analyses

The chart on the next pages describes a set of changes and enhancements to the JCATS model proposed by IDA to improve the model's capabilities for representing combat in a MOUT environment. Many of these changes were proposed through the auspices of the MOUT ACTD, based on experiences and lessons learned during the course of this program. These proposals were all made independent of the MOUT V&V effort described in this report. They are merely presented here to provide an historical record.

	A	B	C
1	**Enhancements to Improve JCATS' Representation of MOUT**		
2	7-May-01		
3			
4	**Enhancements (previously discussed)**	**Notes**	**Status**
5	Human behaviors model		done
6	See-through holes made in walls (a system can see through a wall that has been breached)		done
7	Engineer obstacles can be placed inside buildings		done
8	Systems (e.g., a robot) can be subjected to an "electronic" kill		done
9	Robot mobility class	priority as of Sept. 2000 meeting	done
10	Trigger nodes to assist planning process (e.g., turn "shoot on" at this point of movement path)	priority as of Sept. 2000 meeting	done
11	Path creation tool: Creates a track that any entity can follow – just need to position the system on the start of the path, and it will follow. Will be good for patrol routes.	priority as of Sept. 2000 meeting	done
12	Representation of a through-wall sensor	priority as of Sept. 2000 meeting	done
13	Ability to ascend the exterior of a building, ladders, fire-escapes	priority as of Sept. 2000 meeting	
14	Ability to create a hole in a wall using a munition, rather than breaching	priority as of Sept. 2000 meeting	
15	Laser designator that can be associated with area directed fire	priority as of Sept. 2000 meeting	
16	Represent burning of buildings (in addition to a preliminary work-around)	Too difficult. How to model the actions of the people inside moving toward the doors? How to model the restriction of a floor? Need experts and a supercomputer to come up with anything realistic.	

	A	B	C
17	Create holes of any size and location in walls and floors dynamically.	Breaching has changed. Now a system can breach a hole first then move (in previous versions, breaching basically was moving through a wall in a specified period of time). A system can just look out of the hole rather than moving through it (version 2.2). There is no capability to breach through floors.	
18	Represent stairs and stairwells both internally and externally to buildings.	No stairs/stairwells. There are questions about whether or not to model individual steps (for JCATS integration with something like DI Guy). Also question of how to build the terrain. There are ramps in JCATS, but those are external.	
19	See and shoot through holes made in walls and floors.	Yes, can see and shoot through holes in walls, but not in floors.	
20	Ascend stairwells and ladders at various speeds.	No	
21	Create objects such as furniture inside buildings.	No furniture but can create engineer obstacles and unattended sensors inside buildings. Could create generic cover & concealment areas inside.	
22	Create air ducts and crawl space inside buildings.	No. Can create varying floor heights.	
23	Create fractional height walls such as counters inside buildings.	Can create a transparent window (of specified size) within an opaque wall.	
24	Designate carpeted areas or other areas that may block robot movement.	No trafficability inside buildings. Intend to create a trafficability class that will allow systems to have different experiences on the same kind of terrain.	
25	Create dark rooms that may be illuminated no matter what the light conditions outdoors.	No. What degree of lighting is needed? There were problems with pitch-black stairways during Urban Warrior. Maybe some pre-calculated light in outer rooms with windows – give them some percentage of light while the inside rooms are dark? Right now there is uniform lighting in buildings. Can have a uniformly light bldg in the nighttime but not a dark building in the day time. However, could have a night scenario and put area lights between the buildings. Also, could create area lights inside a bldg to represent different lighting in individual rooms.	

	A	B	C
26	Employ stand-off breaching means.	No. Chris explained to LLNL how this is one of the MOUT ACTD requirements. See comment below about Chris's idea for creating a list of technologies for LLNL. DBBL used a work-around. Mounted an "RLEM" system on a soldier. That RLEM went up to the door at very high speed and breached it, which kept the soldiers from being exposed during breaching.	

	A	B	C
27	**Enhancements to Improve Robot Representation (previously discussed)**	**Notes**	**Status**
28	Tethered to a person, concept of ownership		
29	Ability to represent different sized robots with very different trafficabilities		
30	Represent one robot throwing another?		
31	Represent a throwbot being thrown into an upper story window	Getting the robot into the window will be the problem. Maybe dismount and let it ascend the building at very fast speed?	
32	Create a means for turning an entity's sensors off when its energy level reaches zero	With robot class	
33	Create a means for keeping entities within a fixed radius of one another (e.g., robots and their human operators)	With robot class	
34	Designating electrical energy as energy source, e.g., batteries, etc.	Currently modeled as fuel. There is no separate representation of energy drain due to mobility and sensor packages. Could use analogy	
35	Allowing movement inside and outside, into and out of buildings.	On first floor only. Should just model a robot that needs to go to upper stories as a person rather than as a vehicle.	
36	Permitting carrying non-lethal weapons.	Yes, but only for planned direct fire, not for automatic fire. Can model non-lethal mines.	
37	Impede movement of small robots at curbs, and permit climbing of curbs.		
38	Subjecting robot sensor packages to electronic "kill".	Yes. And can model electronic suppression too. Beam weapon	
39	Automating certain actions, e.g., seeking cover to avoid detection.	Working on it – see notes at end of document about the "state machine"	
40	Allowing movement of robots in formations, including "follow the leader" formations where one robot follows another	Yes they can move in aggregation, but not in a leader-follower type setup like GA Tech had in mind.	
41	Including anti-handling devices for robots (e.g. devices to shock individuals who try to pick up robots)	No. As a work-around could use short-range weapon on robot with suppression characteristics.	
42			

	A	B	C
43	**Other Enhancements**	**Notes**	**Status**
44	Ability to plan area-of-fire using an x-z representation of the building (rather than birds-eye view like we have now)		
45	Ability to "mark" individuals--to vary by persistence, range of visibility of the mark, range of application of the mark		
46	Lights mounted on people (flashlights)		
47	Behavior model applicable to systems other than dismounted systems		
48	Lethal kills by beam weapon		
49	More control over smoke effects (e.g. time to fill a volume, volume filled, time to decay, density, etc)--useful for smoke between buildings		
50	Some representation of smoke inside buildings		
51	Ability to read movement track data from real systems into JCATS for replay/further study. Could be useful in reading in movment tracks from an instrumented site like McKenna MOUT site.		
52	Sniper detection device (varies by the number of shots required before the sniper's detection can be predicted; accuracy of sniper location--window, building, block; range)		
53	Representation of thermobaric and other overpressure weapons		
54	Crowd behavior representation (based on intelligent agent research)		
55	Body armor (ability to modify PH/PK by allowing soldier to wear or shed body armor)		
56	Concept of ownership for devices like ladders, breaching devices		
57	Improved Acoustic model (e.g. directional sensor)		
58	Representation of parachute/parafoil for delivery of individual troops, systems, equipment, logistics		
59	Representation of active protection systems for vehicles		
60	Representation of batteries/battery usage for sensors (stand-alone and on other systems)		
61	Representation of magnetic detectors (magnetometers)		
62	Representation of (disruptable) linkage (physical or electromagnetic) between stand-alone sensors and other weapon systems/individual troops		

APPENDIX K

JCATS VALIDATION SCENARIO DESCRIPTIONS

Institute for Defense Analyses

WN13: Improved Forcible Entry

Capabilities
a) Infantry forcible entry into buildings
- Explosive
- KE
- Mechanical
- Directed Energy
- Chemical
b) Vehicle breach of walls
- Explosive
- KE
- Mechanical
- Directed Energy
- Chemical
c) Vehicle ability to clear/reduce obstacles
- Explosive
- KE
- Mechanical
- Directed Energy
- Chemical

General approach: JCATS represents both wall and obstacle (wire, sandbags, hulks, rubble) breaching. DBBL has already modeled stand-off and conventional forms of breaching for the MOUT ACTD. In their modeling of the selected suite of requirements, DBBL modeled wall-breaching devices, Rifle-Launched Entry Munitions (RLEM) and door-breaching devices. They later modeled RLEM again in the aggregate force effectiveness study.

MOUT V&V considerations: 13a. Infantry forcible entry into buildings - simulate dismounted troops breaching and entering the first floor of a building.
13c. Vehicle ability to clear/reduce obstacles –simulate a vehicle clearing an obstacle in the street and then securing a street. Red would defend the street.

Hypothesis: None of the capabilities listed within each subneed provides better improvements to force effectiveness than any other capability listed for that same subneed.

Does this need require gaming: Yes. Since DBBL already has a scenario and has modeled some capabilities, they should continue their work in this area and game the capabilities to fill the sub-needs above.

Scenario outline: DBBL will define appropriate scenario

Assumptions:

Measures: Ammunition Expenditure, FER, LER, Red losses, Blue losses, Time for Sub Units to Move Between Critical Nodes, Time to Accomplish Mission.

Experimental design:

Data requirements: The different capabilities can be represented according to whether or not the capability is stand-off, the range at which it operates, the time required to breach, the size of the opening created, etc.

JCATS specific inputs:

Walls
- Breach time (sec)
- Breach size (width)

Wire/rubble/mines
- Speed (km/hr) of movement through obstacle
- Size of breach is the width of the vehicle

WN15: Knowledge of Other Side of Wall

Capabilities
- Through-wall sensing
- Robotics
- Physical Penetration

General approach: The approach for this need will be to model through-wall sensors, robotics, and physical penetration as ways for individuals to get information about people or activities on the other side of a wall. Robotics include both UAVs and UGVs and we may want to model both. Physical penetration could mean breaching a small hole in the wall or sliding something beneath the wall/door.

MOUT V&V considerations: Perform floor clearing operation using through-wall sensors, robotics, and physical penetration. Red would defend the building.

Hypothesis: No capability listed above provides better increases in force effectiveness than any of the other capabilities.

Does this need require gaming: Yes

Scenario outline: This scenario will be determined by DBBL.

Assumptions:

Measures: Critical Items/Activities Detected, Non-combatants Detected, Red Targets Acquired by Blue, Ammunition Expenditure, FER, LER, Red Losses, Blue Losses – fratricide, Blue losses (by red), Blue Target Detected/Acquired by Red, Non-combatant

losses, Time for Sub-Units to Move Between Critical Nodes, Time to Accomplish Unit Mission

Experimental design:

Data requirements: For all three capabilities, the modelers will need to know what types of information the capability provides. Possibilities include: occupied vs. unoccupied, number of people in the room, whether or not there are weapons in the room, information about where the people are located in the room, a camera-view of the room, whether or not there are enemy in the room, etc.

- Through-wall sensor – range, the number of walls it can penetrate, whether or not it is man-portable, the time required to achieve detections in the next room, and of course the type of information provided.
- Robotics (to include UGVs and UAVs) – dimensions, types of sensors on the robot, any weapons that the robot may carry, speed of movement, whether or not the robot is tele-operated (and if so, the range at which the robot is tethered to the operator and whether LOS is required), whether or not the robot can breach doors, is tall enough to look into windows, etc. The modelers also need to know what kind of information the robot gathers. Is the robot's camera view shown to the operator, or does the robot have some kind of metal detecting device mounted on it, etc. Is the robot loud – will it alert the enemy of its presence, or is it silent?
- Physical penetration – time to penetrate (whether that means sliding something under the door, or creating a hole in the door), type of information is gained, how much of a distraction is presented to the enemy.

JCATS specific inputs:

Through-wall sensor

- Min/max range (m)
- FOV (degrees)
- Acquisition scan interval (sec)
- Probability of detection/scan interval
- Whether or not the sensor is electronic
- Maximum concurrent acquisitions (#)
- Reliability (%)
- Detect only moving entities
- Detect only dismounted entities
- Limited by X number of walls (#)

Additional modeling for through-wall sensor study: We need to have discussions with the sponsor before we can plan this study. However, some questions which we may want to investigate include analyzing the benefits that the different types of information provide. Is it important to know where the people are in the adjacent room, or just to know that the room is occupied? Scenarios could be gamed where the JCATS operators are provided

with different levels of information and are then allowed to script their reactions based on that information.

WN18: Get on Top of Buildings

Capabilities
- Mechanical
- Propulsion
- Explosive
- Aerial

General approach: Mechanical, propulsion, and explosive capabilities will all differ by the amount of time required to get a person to the top of a building, the setup time, availability (BOI/ownership), height reachable, and protection provided. Aerial delivery provides another option – perhaps a helicopter delivers the person.

These capabilities could be modeled as a person moving between floors using a "go to floor" node in a clear-walled addition to the building. Alternatively, we could probably model the people as small helicopters (or mount the people on person-size helicopters) to ascend the outside of the building.

DBBL has already modeled ladders as a part of the aggregate force effectiveness study for the MOUT ACTD. We recommend that they use that scenario to model the other capabilities, like helicopters, that could be used to get people to the top of buildings.

MOUT V&V considerations: Conduct an ambush in a street. Could use armored vehicles (APV). May be fired from building (roofs and windows).

Hypothesis: No capability listed above provides any more improvement to force effectiveness than any of the other capabilities.

Does this need require gaming: Yes

Scenario outline: The scenario will be developed by DBBL

Assumptions:

Measures: FER, LER, Red Losses, Blue Losses, Time to Accomplish Unit Mission

Experimental design:

Data requirements: Speed of ascent, time to prepare, height achievable, basis of issue/ownership, loudness (so that the enemy can react appropriately)

WN19: Enhanced Indirect Fires

Capabilities
- Improvements to existing mortars
- Accuracy
- Variable effects

General approach: Would more accurate mortars or more control over the effects of the mortars provide more of an increase in force effectiveness? Variable effects could be modeled by using three (or some other number of) different types of mortar munitions associated with the mortar weapon. The individuals operating JCATS could choose to fire the appropriate one.

Hypothesis: No indirect fire capability listed above provides better force effectiveness to the side using the capability than any of the other capabilities listed above.

MOUT V&V considerations: Attack a bunker. Red would defend against attack.

Does this need require gaming: Yes

Scenario outline: To be determined by DBBL

This scenario will also be built from the McKenna room-clearing scenario. There will be an enemy squad/fire team placed just below/south of the building being cleared by the friendly forces. The enemy squad will fire the improved mortars on the friendly forces clearing and waiting beside the building.

Assumptions:

Measures: Ammunition expenditure, Average engagement ranges, FER, LER, Red losses, Blue losses, Non-combatant losses, Time to Accomplish Mission.

Experimental design:

Data requirements:

JCATS specific inputs:

Accuracy—

- In range data, change Aiming Error Deflection and Range to values near zero
- In range data, change Ballistic Error Deflection and Range to values near zero

Variable HE effects—

- Burst height (m)
- Lethality angles (at 1/3, 2/3, and max range) (degrees)
- Lethal area

APPENDIX L

JCATS VALIDATION QUESTIONNAIRES

Institute for Defense Analyses

These questionnaires were developed to facilitate the validation process for JCATS[1]. The purpose of this validation effort was to assess whether JCATS approximates the real world MOUT to the greatest fidelity possible. This effort was accomplished by using subject matter experts (SMEs) with knowledge of, and familiarity with, urban operations who were asked to provide insights and judgments on how well JCATS represents "real" combat. These experts included individuals with considerable experience conducting and observing JCATS gaming, such as the personnel at Fort Benning Simulation Center, and individuals with considerable experience conducting and observing urban training exercises and who also have been involved in actual U.S. military operations in urban environments. The knowledge and experience of a select group of such people can be used to isolate and focus on key elements of urban combat. These elements can be represented in the model, and the SMEs asked to make judgments both on the operations as they take place on the JCATS screen as well as on the model's processed output.

The validation process addresses how well JCATS represents MOUT and MOUT activities. Operational SMEs should evaluate the following statements and questions using a 1 to 5 rating (5 very well and 1 not at all).

1) Does JCATS produce results that are feasible?
   1 2 3 4 5
   Comments:

2) Does a difference in the input produce the expected proportional change in the output?
   1 2 3 4 5
   Comments:

3) Do the levels of force structure and interaction have sufficient fidelity and resolution?
   1 2 3 4 5
   Comments:

4) Based on your military experience, does JCATS compare favorably to historical, test, laboratory, and/or exercise data?
   1 2 3 4 5
   Comments:

[1] The Joint Non-Lethal Weapons Directorate conducted a V&V that was completed in October of 2000. The questions and statements used for the validation portion of that effort were used as a starting point for these questionnaires.

5) Does JCATS adequately represent a MOUT environment?
1 2 3 4 5
Comments:

6) Is JCATS suitable for the overall intended use as an analytical tool?
1 2 3 4 5
Comments:

Structural validation focuses on the internal structure of JCATS in the context of its intended use. Programmers should answer the following questions using a 1 to 5 rating (5 very well and 1 not at all).

1) Is JCATS sensitive to the data input values?
1 2 3 4 5
Comments:

2) Does JCATS adequately represent the real world?
1 2 3 4 5
Comments:

3) Is JCATS complete and are the functions adequately modeled?
1 2 3 4 5
Comments:

4) Is there adequate and consistent representation of terrain and environment across all JCATS components?
1 2 3 4 5
Comments:

5) Can JCATS output/results be used clearly, adequately, and appropriately to address MOUT problems?
1 2 3 4 5
Comments:

6) Can JCATS runs be accomplished and results analyzed in a timely manner?
1 2 3 4 5
Comments:

7) Are baseline scenarios, terrain data, threat data, and weapon performance data for JCATS database available?
1 2 3 4 5
Comments:

8) Are terrain and environment representations functionally adequate to address MOUT issues?

1 2 3 4 5
Comments:

9) Are the clarity, fidelity, complexity, and level of detail of the simulated entities acceptable for its intended usage?
1 2 3 4 5
Comments:

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE (DD-MM-YY)	2. REPORT TYPE	3. DATES COVERED (From – To)
October 2002	Final	

4. TITLE AND SUBTITLE
JCATS verification and Validation Report

5a. CONTRACT NO. DASW01-98-C-0067

5b. GRANT NO.

5c. PROGRAM ELEMENT NO(S).

6. AUTHOR(S)
W. M. Christenson, Mary Catherine Flythe, Terri J. Walsh, Robert A. Zirkle

5d. PROJECT NO.

5e. TASK NO. AJ-6-1543

5f. WORK UNIT NO.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311-1882

8. PERFORMING ORGANIZATION REPORT NO.
IDA Document D-2791

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of the Deputy Under Secretary of Defense
for Advanced Systems and Concepts
3700 Defense Pentagon
Washington, DC 20301-3700

10. SPONSOR'S / MONITOR'S ACRONYM(S)
ODUSD(AS&C)

11. SPONSOR'S / MONITOR'S REPORT NO(S).

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This paper asserts that JCATS best represents combat in urban areas, and comes closest to meeting the MOUT capability requirements. However, before JCATS could be fully utilized for MOUT analysis purposes, both the model's urban combat representation and the relevant database needed to be subjected to appropriate verification and validation efforts. This paper documents the results of an effort to undertake the first of these requirements: a V&V, the model's representation of combat in the urban environment.

15. SUBJECT TERMS
V&V, Verification, Validation, MOUT, JCATS, Modeling and Simulation, force on force modeling

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NO. OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	220	Mr. Robert Shields
U	U	U			19b. TELEPHONE NUMBER (Include Area Code) (703) 695-9873